Cimatron

Cimatron
CAD/CAM Solutions for Manufacturing

Data Interface Utilities

Version 12

Data DI ▸ DFM ▸ NC Steel

All rights reserved by Cimatron Ltd.

No part of this software or document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying or recording, for any purpose, without written permission from Cimatron Ltd.

Cimatron may make improvements and changes in the software described in this document at any time and without prior notice. These changes will be documented in future editions of this publication.

© Copyright Cimatron Ltd. 1984-2001

Cimatron, Cimatron[it], Cimulator, Sketcher, CimaRender, CimaDEK, MoldBase, MoldBase 3D, Re-Enge, CPDM, MPDM, Cimagrafi and the C logo design are trademarks of Cimatron Ltd. QuickCompare, QuickElectrode, QuickConcept, QuickSplit, QuickMold and Quick Tooling are pending trademarks of Cimatron Ltd. Cimatron, Cimulator, Sketcher, CimaRender, CimaRender Pro, CimaDEK, Cimagrafi, and the C logo design are registered in the U.S. Patent and Trademark Office. CimaRender and CimaRender Pro are copyrighted by Graffiti Software Industries Ltd. MoldBase 3D is based on a product copyrighted by R&B Ltd. Cimatron IMSpost is based on a product copyrighted by Intelligent Manufacturing Software, Inc. IMSpost is a trademark of Intelligent Manufacturing Software, Inc. MODView is copyrighted by Dataface, Co. Ltd.

All other company and product names are trademarks or registered trademarks of their respective owners.

All references to other trademarks and/or copyrighted material is for identification purposes and/or unintentional.

Cimatron Ltd. is not necessarily associated with any other product or vendors mentioned herein.

Disclaimer of Warranty and Liability.

> No representations or warranties, expressed or implied, of any kind are made by or with respect to anything in this document.
>
> In no event shall Cimatron Ltd., its employees or previous employees, be liable for any incidental, direct or indirect, special or consequential damages whatsoever (including but not limited to loss of profits) arising out of or related to this manual, and/or the product, or any use thereof.

Revised 2001.

Preface

Cimatron develops, markets and supports tools to automate the mechanical engineering process. Our systems support all phases of product development, with solutions for computer aided design (CAD) and manufacturing (CAM). **Cimatron**'s *integrated technology* approach combines design tools with optimized command output to computer-controlled manufacturing equipment. Drafting-table-to-shopfloor integration lets Cimatron clients realize dramatic efficiencies in product development and manufacturing.

Cimatron^{it} - Cimatron's flagship product - covers the entire spectrum of design, engineering and manufacturing processes, including:

- A complete range of **wireframe**, **surface** and **parametric solid modeling tools** with rendering capabilities;
- Advanced **assembly**, **sub-assembly** and **part management**, and **associative drafting functionality**;
- Comprehensive, accurate **data exchange interface utilities** covering **DXF**, **DWG**, **IGES**, **JAMA-IS**, **VDA**, **PTC**, **STEP**, **SAT**, **CATIA** and **UNIGRAPHICS**;
- Powerful and intelligent **NC** applications for precise multi-axis machining.

The modular yet integrated structure of Cimatron^{it} grows to accommodate cutting edge tools and techniques. These now include the new **Quick Tooling** applications:

- **QuickSplit**
  QuickSplit automates the search and separation of core, cavity and sliders to assist in determining the number of actions required to create a mold. After separating core, cavity and slides, QuickSplit identifies the parting lines and generates the parting surface.
  Automatic and interactive tools allow the construction of parting surfaces for any complex geometry. Embedded Draft Analysis enables designers to identify potential problems with undercuts and confirm minimum draft per side.
  QuickSplit is tolerant of surface models with gaps, mismatched boundaries or missing faces, therefore bypassing data corrections and saving precious time.
  Component motion animation, dynamic cross-sectioning and clipping planes, reduce human error and verify parting design. QuickSplit enables several trial and error iterations in a very short time - resulting in optimal draw directions.
- **QuickElectrode**
  QuickElectrode is an EDM electrode design solution used for shortening the electrode process. QuickElectrode is used for burn area selection, electrode design, management, documentation and manufacturing.
  The QuickElectrode Navigator enables full control over the display and activation of electrodes, while allowing several users to collaborate on the same part.

QuickElectrode's report generation features includes set-up sheets, burn location reports and a full electrode schedule, therby alleviating the tedious task of documenting the process .

- **QuickConcept**
  QuickConcept is a preliminary design and review package which allows tool designers and their suppliers to hold *virtual* review meetings over the Internet in real-time. Multiple users can connect to each other to section, label, dimension, and identify points of interest and problem areas of any given tool. All members of the review meeting will interactively view the same screen at the same time.

- **QuickCompare**
  QuickCompare assists the tool designer in determining the scope and effect of Engineering Changes (ECOs) on the tooling process. QuickCompare mathematically compares the geometrical differences between two sets, graphically marks these differences and documents the changes in a CAD file. Here, the designer updates related components and tooling, while archiving ECOs. The typically long *CAD investigation process* is significantly shortened. QuickCompare ensures that all ECOs have been located, whether or not they were communicated from design.

- **MoldBase 3D**
  MoldBase3D offers an innovative *wizard-based* approach to parametric mold base design. MoldBase3D automatically creates 3D solid (parametric & associative) moldbases, with all components and accessories, from industry-standard catalog suppliers such as HASCO, DME, PCS, FUTABA, DMS, PEDROTTI, RADOURDIN, SIDECO, STRACK and MISUMI. Creation of the assembly and detailed drawings of each plate are automated, complete with 2D and 3D section views, ordinate dimensions, labels, balloons, and an itemized Bill of Materials. This module is fully associative to the mold design and changes are automatically reflected in all stages of the design process.

Cimatron's automated engineering expertise benefits many industries, as competition requires tighter development cycles and efficient fabrication.

Powerful modules within Cimatron^{it} expand your system's capabilities. These may be purchased from your Cimatron representative.

This publication provides a detailed description of the major features of the appropriate Cimatron^{it} application/topic. It is intended to help users in the daily operation of Cimatron^{it}.

A list of Cimatron^{it} documentation, for the current version, is shown on the next page.

Cimatron Documentation

Cimatron[it] documentation comprises Reference Manuals, On-Line Help and Tutorials which together provide a comprehensive guide to Cimatron[it].

The list of Cimatron[it] documentation, for the current version, is as follows:

Cimatron[it] Reference Manuals

Publication	Description	Display Options *
Fundamentals & General Functions	Introduction to the fundamentals of Cimatron[it] and description of the General functions.	A H
Modeling	Description of the wireframe and surface Modeling functions.	A H
QuickSplit	QuickSplit automates the search and separation of core, cavity and sliders to assist in determining the number of actions required to create a mold.	A H
QuickElectrode	QuickElectrode is an EDM electrode design solution used for shortening the electrode process.	A H
QuickCompare	QuickCompare mathematically compares the geometrical differences between two models, graphically marks these differences and documents the changes in a CAD file.	A H
Drafting	Description of the Drafting functions.	A H
Solid Modeling	Solid Modeling functions including Sketcher.	A H
MoldBase 3D	Description of the functions associated with the detailed design of mold plates and components. MoldBase3D offers an innovative *wizard-based* approach to parametric mold base design.	A W
Numerical Control	Description of the NC functions.	A H
Cimatron IMSpost	Cimatron IMSpost is a macro-based system for developing and customizing postprocessors.	A W
General Post Processor	General Post Processor (GPP) functions.	A
Finite Element Modeling	Description of Finite Element Modeling (FEM) functions.	A H
Utilities	Various utilities that may be used with Cimatron[it]. These utilities are either **Internal**, run via the USER function, or **External**, run via the Main Menu.	A H
Data Interface Utilities	Description of Cimatron's *comprehensive* data interface utilities; DXF, DWG, IGES, JAMA-IS, VDA, PTC, STEP, SAT, CATIA and UNIGRAPHICS.	A W
CimaDEK	Cimatron's specialized Developer's kit, for programming customized functions.	A
CimaRender Pro	A photo-realistic rendering package.	A W
MPDM: Getting Started **MPDM: Administrator**	Description of how to use Manufacturing Product Data Management to track and organize all files and data associated with a project.	A W
Re-Enge	Description of Reverse Engineering design functions.	A

Cimatron[it] Tutorials

Design - covers **QuickCompare** and **QuickElectrode**.	A H
Drafting - covers the DMS function.	A H
NC - covers the differences between versions 11 and 12.	A H

*** Legend**:

A	Acrobat PDF
H	HTML
W	Winhelp

❑

Table of Contents

Introduction

About This Manual. Intro-1
Typographical Conventions. Intro-4

Chapter 1 Introduction to Data Interfaces

Running the Data Interface Utilities 1-1
Interaction - Graphic 1-2

Chapter 2 Introduction to DXF and DWG Utilities

Running DXF 2-2
Graphic 2-2
Textual 2-2
About the Drawing Interchange File (DXF). 2-3
What is DXF? 2-3
How Does it Work? 2-3
Overview of the DXF File Format 2-4
DXF ASCII File Format 2-4
DXF File Syntax 2-5
Group Code Assignment 2-5
HEADER Section 2-5
TABLES Section 2-6
BLOCKS Section 2-7
ENTITIES Section 2-8

Chapter 3 The Read-DXF Utility

Running Read-DXF 3-2
Graphic 3-2
Textual 3-2
Running Modes 3-3
Interactive 3-3
Batch 3-3
Graphic Interaction with Read-DXF 3-4
Files 3-5
Params 3-8
Mapping 3-9
Textual Interaction with Read-DXF 3-11
Attribute Mapping 3-12
Entity Mapping 3-13

Chapter 4 The Write-DXF Utility

Running Write-DXF 4-2
Graphic 4-2
Textual 4-2
Running Modes 4-3
Interactive 4-3
Batch 4-3
Graphic Interaction with Write-DXF 4-4
Files 4-5
Params 4-9
Filter 4-10
Mapping 4-11
Textual Interaction with Write-DXF 4-12
Application Mapping 4-13
Attribute Mapping 4-13
Entity Mapping 4-14
Constraints 4-15

Chapter 5 The DXF Utilities Parameter File

Parameter File Name 5-1
Parameter File Creation 5-1
Parameter File Format and Rules 5-2
Parameter File Statements and Syntax 5-2
Read-DXF Parameters 5-3
Operation Mode Parameters 5-3
Statements for Conversion 5-4
Statements for Text Conversion 5-5
Write-DXF Parameters 5-7
Operation Mode Parameters 5-7
Additional Statements 5-8
Statements for Text Conversion 5-10
Selective Translation of Cimatron Files to DXF Files 5-11
User Input 5-11
Filter Syntax 5-11
Application Filters 5-12
Attribute Filters 5-14
Logical Operation Rules 5-15
Number of Precision Digits 5-16
Statements for Conversion 5-16
Selective Translation Statements for Write-DXF Utility 5-16
USE FILE Statement - Nested Parameter File 5-17
Default Parameter File and Parameter Values 5-17
Parameter File Check 5-18

Chapter 6 DXF File Information Utility

Running the DXF File Information Utility 6-1
Graphic 6-1
Textual 6-1
Sample DXF Report File 6-2

Chapter 7 PFM File Information Utility

Running the PFM File Information Utility 7-1
Graphic 7-1
Textual 7-1

Chapter 8 Introduction to the IGES Utilities

Running IGES 8-2
Graphic 8-2
Textual 8-2
About the Initial Graphics Exchange Specification (IGES) 8-3
What is IGES? 8-3
How Does it Work? 8-3
Overview of the IGES File Format 8-4
IGES File Structure 8-5
Start Section 8-6
Global Section 8-6
Directory Entry Section 8-10
Parameter Data Section 8-10
Terminate Section 8-11

Chapter 9 The Read-IGES Utility

Running Read-IGES 9-2
Graphic 9-2
Textual 9-2
Running Modes 9-3
Interactive 9-3
Batch 9-3
Graphic Interaction with Read-IGES 9-4
Files 9-5
Data 9-8
Params 9-9
Filter 9-11
Mapping 9-13

Textual Interaction with Read-IGES 9-14
Entity Mapping 9-15
Geometric Entities 9-15
Annotation Entities 9-16
Structure Entities 9-17

Chapter 10 The Write-IGES Utility

Running Write-IGES 10-2
Graphic 10-2
Textual 10-2
Running Modes 10-3
Interactive 10-3
Batch 10-3
Graphic Interaction with Write-IGES 10-4
Files 10-5
Data 10-9
Params 10-10
Filter 10-12
Mapping 10-14
Textual Interaction with Write-IGES 10-15
Entity Mapping 10-16
Geometric Entities 10-16
Annotation Entities 10-18
Structure Entities 10-18
Constraints 10-19

Chapter 11 The IGES Utilities Parameter File

Parameter File Name 11-1
Parameter File Creation 11-1
Parameter File Format and Rules 11-2
Parameter File Statements and Syntax 11-2
Read-IGES Parameters 11-3
Operation Mode Parameters 11-3
Statements for Text Conversion 11-9
Statement for Line-Font Conversion 11-10
Read-IGES Utility Interaction Statements 11-10
Additional Read-IGES Utility Statements 11-10
Write-IGES Parameters 11-11
Operation Mode Parameters 11-11
Additional Statements 11-12
Statements for Text Conversion 11-17
Statement for Line-Font Conversion 11-18
Additional Write-IGES Utility Statements 11-18
Statements for Text Conversion 11-19
Selective Translation Statements for Read-IGES Utility 11-19
Selective Translation Statements for Write-IGES Utility 11-20
Write-IGES Utility Interaction Statements 11-21
Logical Operation Rules 11-22
Clarifications and Constraints 11-23
USE FILE Statement - Nested Parameter File 11-23
Selective Translation of Cimatron Files to IGES Files 11-24
User Input 11-24
Filter Syntax 11-24
Application Filters 11-25
Attribute Filters 11-27
Selective Translation for Read-IGES Utility 11-28
Attribute Filters 11-28
Entity Filter 11-29
Default Parameter File and Parameter Values 11-29
Parameter File Check 11-30

Chapter 12 IGES File Information Utility

Running the IGES File Information Utility. 12-1
Graphic 12-1
Textual 12-1
Sample IGES Report File 12-2

Chapter 13 PFM File Information Utility

Running the PFM File Information Utility 13-1
Graphic 13-1
Textual 13-1

Chapter 14 Introduction to the PTC Utilities

Running PTC 14-2
Graphic 14-2
Textual 14-2
About the PTC (Neutral Files) 14-3
Overview of the PTC (Neutral Files) 14-3

Chapter 15 The Read-PTC Utility

Running Read-PTC 15-2
Graphic 15-2
Textual 15-2
Running Modes. 15-3
Interactive 15-3
Batch 15-3
Graphic Interaction with Read-PTC 15-4
Files 15-5
Params 15-8
Textual Interaction with Read-PTC 15-9

Chapter 16 The PTC Utilities Parameter File

Parameter File Name 16-1
Parameter File Creation 16-1
Parameter File Format and Rules 16-2
Read-PTC Utility Interaction Statements 16-3
Read-PTC Parameters 16-3
Operation Mode Parameters 16-3
Default Parameter File and Parameter Values 16-5
Customized Default Parameter File 16-5
Parameter File Check 16-5

Chapter 17 Introduction to the VDA Utilities

Running VDA 17-2
Graphic 17-2
Textual 17-2
RDVDA 17-3
About the VDA-Interface 17-3

Chapter 18 The Read-VDA Utility

Running Read-VDA 18-2
Graphic 18-2
Textual 18-2
Running Modes 18-3
Interactive 18-3
Batch 18-3
Graphic Interaction with Read-VDA 18-5
Files 18-6
Params 18-9
Textual Interaction with Read-VDA 18-10

Chapter 19 The Write-VDA Utility

Running Write-VDA 19-2
Graphic 19-2
Textual 19-2
Running Modes 19-3
Interactive 19-3
Batch 19-3
Graphic Interaction with Write-VDA 19-4
Files 19-5
Data 19-8
Params 19-9
Textual Interaction with Write-VDA 19-10

Chapter 20 The VDA Utilities Parameter File

Parameter File Name 20-1
Parameter File Creation 20-1
Parameter File Format and Rules 20-2
Read-VDA Parameters 20-3
Operation Mode Parameters 20-3
Read-VDA Utility Interaction Statements 20-6
Write-VDA Parameters 20-7
Operation Mode Parameters 20-7
Write-VDA Utility Interaction Statements 20-8
USE FILE Statement - Nested Parameter File 20-8
Default Parameter File and Parameter Values 20-9
Parameter File Check 20-9

Chapter 21 Introduction to the JAMA-IS Utilities

Running JAMA-IS 21-2
Graphic 21-2
Textual 21-2
About the JAMA-IS Specification 21-3

Chapter 22 JAMA-IS Options

JAMA-IS File Structure 22-2

Chapter 23 Introduction to the STEP Utility

Running STEP 23-2
Graphic 23-2
About STEP 23-3
What is STEP? 23-3
STEP File Structure 23-3
Header Section 23-5
Data Section 23-5

Chapter 24 The Read-STEP Utility

Running Read-STEP 24-1
Graphic 24-1
Textual 24-1
Graphic Interaction with Read-STEP 24-2
Files 24-3
Params 24-6
Entity Mapping 24-7

Chapter 25 The Write-STEP Utility

Running Write-STEP 25-1
Graphic 25-1
Textual 25-1
Graphic Interaction with Write-STEP 25-2
Files 25-3
Params 25-6
Entity Mapping 25-7

Chapter 26 Introduction to the SAT Utility

About SAT 26-1
Running SAT 26-2
Graphic 26-2
Textual 26-2

Chapter 27 The Read-SAT Utility

Running Read-SAT 27-1
Graphic 27-1
Textual 27-1
Running Modes 27-2
Interactive 27-2
Batch 27-2
Graphic Interaction with Read-SAT 27-3
Files 27-3
Params 27-6
Textual Interaction with Read-SAT 27-7

Chapter 28 Introduction to the CATIA Utilities

Running CATIA 28-2
Graphic 28-2
Textual 28-2
About CATIA 28-3

Chapter 29 The Read-CATIA Utility

Running Read-CATIA 29-1
Graphic 29-1
Textual 29-1
Running Modes 29-2
Interactive 29-2
Batch 29-2
Graphic Interaction with Read-CATIA 29-3
Files 29-4
Params 29-7
Filter 29-8
Textual Interaction with Read-CATIA 29-9
Entity Mapping 29-10

Chapter 30 The Write-CATIA Utility

Running Write-CATIA 30-1
Graphic 30-1
Textual 30-1
Running Modes 30-2
Interactive 30-2
Batch 30-2
Graphic Interaction with Write-CATIA 30-3
Files 30-4
Params 30-7
Textual Interaction with Write-CATIA 30-8
Entity Mapping 30-9

Chapter 31 The CATIA Utilities Parameter File

Parameter File Name 31-1
Parameter File Creation 31-1
Parameter File Format and Rules 31-2
Read-CATIA Parameters 31-2
Operation Mode Parameters 31-2
Write-CATIA Parameters 31-4
Operation Mode Parameters 31-4
Parameter File Check 31-4

Chapter 32 Introduction to the Unigraphics Utilities

Running Unigraphics. 32-1
Graphic 32-1
About Unigraphics 32-2

Chapter 33 The Read-Unigraphics Utility

Running Read-Unigraphics 33-1
Graphic 33-1
Running Modes 33-2
Interactive 33-2
Batch 33-2
Graphic Interaction with Read-Unigraphics 33-4
Files 33-5
Params 33-8
Entity Mapping 33-9

Chapter 34 The Unigraphics Utilities Parameter File

Parameter File Name 34-1
Parameter File Creation 34-1
Parameter File Format and Rules 34-2
Read-Unigraphics Parameters 34-2
Operation Mode Parameters 34-2

Appendix A Cimatron File Name Extensions

Cimatron Extensions A-1
DXF Extensions A-1
DWG Extensions A-1
IGES Extensions A-1
PTC Extensions A-1
VDA Extensions A-2
Jama-Is Extensions A-2
STEP Extensions A-2
CATIA Extensions A-2
Unigraphics Extensions A-2

Appendix B DWG Group Codes

Group Codes B-1

Introduction

We recommend that new users of **Cimatron** read at least the first three chapters of the **Fundamentals & General Functions Manual**, to acquire a working knowledge of **Cimatron** software.

As you work with the system, different function overlays will be displayed. At the beginning of many chapters, you will find an overlay diagram which shows the functions described in the chapter and indicates how to access them. Within chapters, functions are listed in alphabetical order. The names of the functions and their options appear in capital letters. A status line at the top of each page tells you which option or sub-option is described on that page.

Although you may only need one particular option within a function, we recommend that you read the description of the interaction for the entire function. Modal parameters, which determine the mode which will be active when the function is executed, are explained in detail for the first option described in each function and briefly in subsequent options.

After you are familiar with the basic system, scan the manual occasionally to discover functions you are not using and to learn how to take full advantage of the power of **Cimatron** software.

About This Manual

This manual provides information on the **Cimatron** Data Interfaces. The chapters of the manual are as follows:

Section I: General

Chapter 1 Contains an overview of the **Cimatron** Data Interfaces and explains how to access them.

Section II: DXF / DWG Utilities

Chapter 2 Contains an overview of the DXF format and the **Cimatron** DXF Utilities.

Chapter 3 Explains the **Read-DXF** utility, and the use of **Read-DXF** parameters.

Chapter 4 Explains the **Write-DXF** utility, and the use of **Write-DXF** parameters.

Chapter 5 Explains the parameters of the DXF Utilities Parameter File and the use of the DXF PARAMETER FILE CHECK utility.

Chapter 6 Explains the DXF FILE INFORMATION utility.

Chapter 7 Explains the PFM FILE INFORMATION utility for the DXF files.

Section III: IGES Utilities

Chapter 8 Contains an overview of the IGES format and the **Cimatron** IGES Utilities.

Chapter 9 Explains the **Read-IGES** Utility, the use of **Read-IGES** parameters and selective translation of IGES files to **Cimatron** files.

Chapter 10 Explains the **Write-IGES** Utility, the use of **Write-IGES** parameters and selective translation of **Cimatron** files to IGES files.

Chapter 11 Explains the parameters of the IGES Utilities Parameter File and the use of the IGES PARAMETER FILE CHECK utility.

Chapter 12 Explains the IGES FILE INFORMATION utility.

Chapter 13 Explains the PFM FILE INFORMATION utility for the IGES file.

Section IV: PTC Utilities

Chapter 14 Contains an overview of the PTC format and the **Cimatron** PTC Utilities.

Chapter 15 Explains the **Read-PTC** utility and the use of the Read-PTC parameters.

Chapter 16 Explains the parameters of the PTC Utilities Parameter file and the use of the PTC PARAMETER FILE CHECK utility.

Section V: VDA Utilities

Chapter 17 Contains an overview of the VDA format, explains the **Cimatron** VDA Utilities, READ VDA and WRITE VDA.

Chapter 18 Explains the **Read-VDA** utility and the use of the **Read-VDA** parameters.

Chapter 19 Explains the **Write-VDA** utility and the use of the **Write-VDA** parameters.

Chapter 20 Explains the **VDA Utility Parameter File**.

Section VI: JAMA-IS Utilities

Chapter 21 Contains an overview of the JAMA-IS format.

Chapter 22 Explains the JAMA-IS options.

Section VII: STEP Utilities

Chapter 23 Contains an overview of the STEP format, explains the **Cimatron** STEP Utilities, READ STEP and WRITE STEP.

Chapter 24 Explains the **Read-STEP** utility and the use of the **Read-STEP** parameters.

Chapter 25 Explains the **Write-STEP** utility and the use of the **Write-STEP** parameters.

Section VIII: SAT Utilities

Chapter 26 Contains an overview of the SAT format.

Chapter 27 Explains the **Read-SAT** utility and the use of the **Read-SAT** parameters.

Section IX: CATIA Utilities

Chapter 28 Contains an overview of the CATIA format.

Chapter 29 Explains the **Read-CATIA** utility and the use of the **Read-CATIA** parameters.

Chapter 30 Explains the **Write-CATIA** utility and the use of the **Write-CATIA** parameters.

Chapter 31 Explains the CATIA Utility Parameter File.

Section X: Unigraphics Utilities

Chapter 32 Contains an overview of the Unigraphics format.

Chapter 33 Explains the **Read-Unigraphics** utility and the use of the **Read-Unigraphics** parameters.

Chapter 34 Explains the Unigraphics Utility Parameter File.

Section XI: Appendices

Appendix A Details the **Cimatron** file name extensions associated with the Data Interface Utilities.

Appendix B Details the DWG Group Codes.

Typographical Conventions

Throughout this manual, certain conventions have been used to present different types of information.

Names

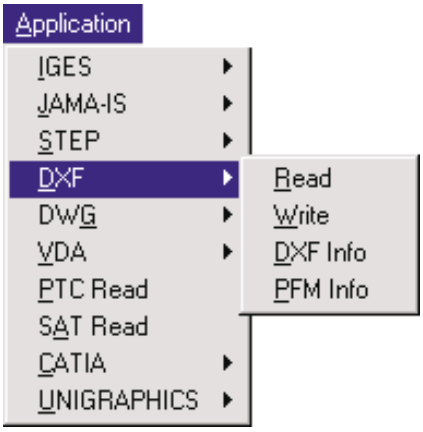
- **Cimatron** function names, standard file names, and file extensions are shown in a bold sans serif font, for example **Read-DWG**, **default.dpr**, **.pfm**.
- **Cimatron** entity names are capitalized, for example Composite Curve.
- DWG and IGES Entity names are capitalized, for example Line.
- IGES Entity names are capitalized and are followed by the type and form numbers of the entity, as defined in the IGES specification, for example Copious Data (Type 106, Form 11).

Actions (optional)

Summarizes the recommended interaction to be followed in the current utility/option, presented in sans serif type. For example:

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select DXF from the Application Menu. The following options are displayed.



3. Select the appropriate option.

Interaction

The prompts which tell you how to execute the function are listed in a lower case sans serif italic letters, and start at the left margin.

Prompt explanations are to the right of the prompts. (Some additional notes are also provided in the right column.) For example:

Replace existing file ? (y/n)

To write over the existing file, type y. To type a different name for the file, type n.

Modal Parameters

Modal parameters are represented in the text by a filled box as shown below. Parameter explanations are to the right of the parameters.

For example:

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

Notes

Provides information to help you avoid problems and achieve accurate results. Each note is preceded by a bullet character for immediate identification. For example:

Note:

- Enter the full path name of the file if it is not located in the current directory.

End of Chapter

The end of a chapter is marked by a box character ❑.

Parameter File Statement

The following conventions are used for presenting a parameter and its arguments.

| - separates different options of the same argument.

{A} - argument is required.

[A] - argument is optional. When the argument is not given, a default program value is used.

<A> - argument is the default value if no value is given.

(n-m) - range of possible values.

Upper case words are fixed predefined keywords or argument values. They should be typed in the parameter file exactly as they appear here.

Lower case words are used for variables for which the user would normally substitute a value. ❑

Section I

General

Chapter 1
Introduction to Data Interfaces

This manual contains descriptions of the Data Interface Utilities available in **Cimatron it**. These data interface utilities convert between the following formats and **Cimatron it**:

- DXF
- DWG
- IGES
- JAMA-IS
- PTC
- VDA
- STEP
- SAT
- CATIA
- UNIGRAPHICS

Each of these data interface utilities is described in a separate section of this manual.

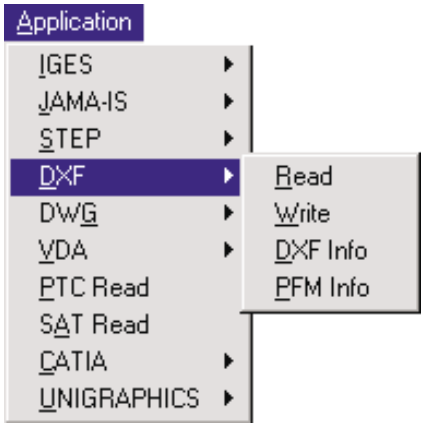
Running the Data Interface Utilities

Two different methods are available for running the Data Interface Utilities, **textual** (type a commands line) or **graphic**. The method that you select and the resulting interaction displayed, are dependent upon the type of hardware and software that you are using, and also upon your preferences.

Hardware	Textual Interface	Graphic Interface
PC: **Windows 95 / Windows 98 / Windows NT**	<application> (from the DOS Shell) or <application> -option <option #> -diprm <DI parameters file name> -difile <input file name> -diout <output file name> -diforce Examples, keyboard: 1. iges: **iges** 2. Write to IGES: **iges -option 2** 3. Read from DWG, use parameter file split.dpr: dwg -option 1 -diprm split	1. double click: **External** icon 2. select the Data Interface tab 3. single click **Data Interface** 4. select: Application from **Application Menu**

Interaction - Graphic

When you select the graphic interaction, the initial dialog box is displayed. Select the appropriate application from the **Application Menu**.



❑

Section II

DXF / DWG Utilities

Chapter 2
Introduction to DXF and DWG Utilities

Part files, including geometric entities, dimensions and kinds of annotation, levels, masters and their instances, may all be transferred between **Cimatron** and other systems in the DXF and DWG formats, developed by Autodesk, Inc.

The DXF and DWG formats essentially contain the same information. DXF files contain ASCII data and DWG files contain binary data. DWG is the most popular format for storing design data; it is the de facto standard.

Due to the similarity of the two formats, the conversion interaction to/from Cimatron is identical. **This section, therefore, applies to both the formats, DXF and DWG.** Wherever DXF is mentioned, it is synonymous with DWG.

The following five utilities are included in the **Cimatron** DXF Utilities:

Read-DXF	Convert files that are in DXF format to files in **Cimatron** part file (**.pfm**) format.
Write-DXF	Convert **Cimatron** part files (**.pfm**) to files in DXF format.
PARAMETER FILE CHECK	Verify a DXF parameter file (**.dpr**).
DXF FILE INFORMATION	Generate a report (.**drt**) on the DXF file tables and entities, and any syntax or structural errors in the file.
PFM FILE INFORMATION	Generate a report (**.pfr**) on the DXF file tables and entities, and any syntax or structural errors in the file.

Running DXF

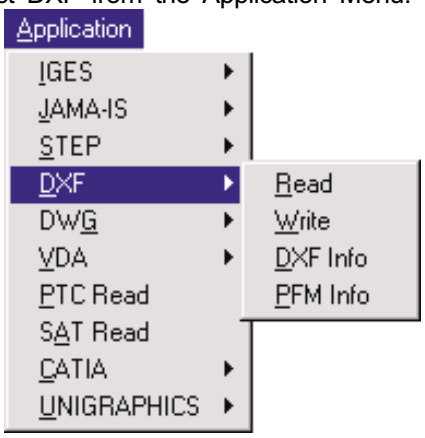
There are two methods of running the DXF Data Interface Utilities:

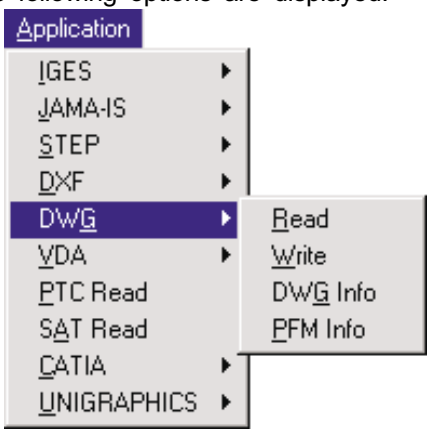
- **graphically**
- **textually**

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select DXF from the Application Menu. The following options are displayed.





3. Select the appropriate option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following Cimatron/DXF options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
4  ->  DXF  File  Information  option
5  ->  PFM  File  Information  option
```

3. Select the appropriate option.

Each of these options will be explained in a chapter of this section.

About the Drawing Interchange File (DXF)

What is DXF?

CAD/CAM systems are in use today in increasing numbers for applications covering all phases of product's design, analysis, manufacturing and testing. As users in each of these stages may use systems supplied by different vendors, there is a need for data to be exchanged digitally between these systems.

DXF, the Drawing Interchange File, answers the need by defining a neutral data format that enables digital exchange of information between AutoCAD and other CAD/CAM systems. All AutoCAD implementations accept this format and are able to convert it to and from their internal drawing file representation.

How Does it Work?

The databases of CAD/CAM systems from different vendors are often incompatible with one another. For example, an arc on one system may be defined by its center, diameter start angle and end angle, while on another system an arc may be defined by a center point, start point and end point. DXF solves this problem by providing a common format for the digital exchange of database information that is independent of all CAD/CAM systems.

Translators convert a proprietary internal database format into a neutral DXF format and from the DXF format into another vendor's internal database. These translators (pre- and post-processors) are normally supplied as part of a vendor's product.

Overview of the DXF File Format

This section contains a quick overview of the DXF format currently supported by **Cimatron**. For more information, refer to the AutoCAD Customization Manual.

Cimatron supports only the ASCII (American National Code for Information Interchange) text file DXF format. This format allows any ASCII text editor to read and modify DXF files.

DXF ASCII File Format

The DXF file contains four sections: HEADER, TABLES, BLOCKS and ENTITIES, and ends with EOF (end of file), as shown in Figure 2-1. Each section begins with a header, and ends with ENDSEC (end of section).

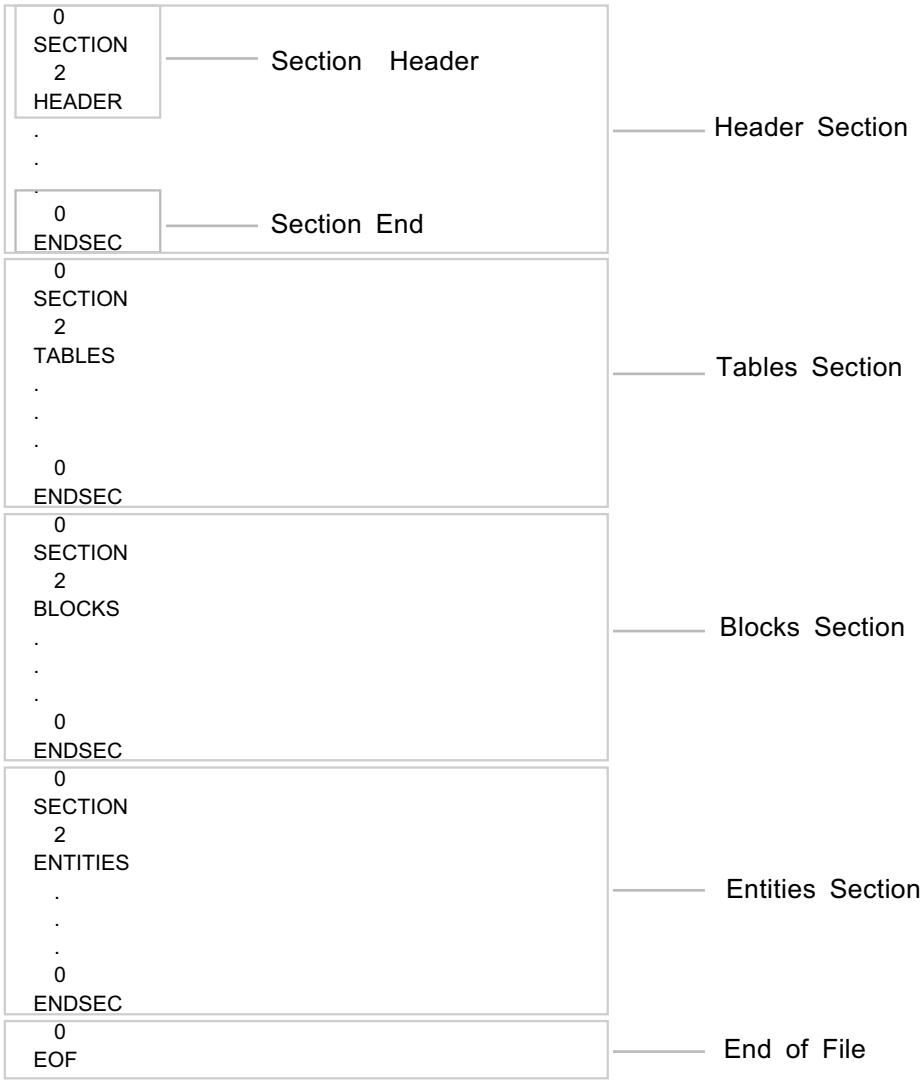


Figure 2-1: Example of ASCII DXF File Format

DXF File Syntax

The DXF file is composed of groups, each of which contains two lines: a group code, followed by a group value.

The group code is a positive integer which specifies the format of the group value. A listing of the group codes is given in Appendix B.

The group value consists of a string of up to 256 characters. For further information about group values derived from group codes, refer to Appendix B.

Group Code Assignment

DXF file items (variables, table entries and entities) are described by a group that introduces the item, followed by sequence of groups that supply information associated with the item.

Group code assignment depends on the item being described. For more information about group code assignment, refer to the AutoCAD Customization Manual.

HEADER Section

The HEADER section of the DXF file contains AutoCAD system variable information associated with the drawing. Each variable in the header section is introduced by the group code 9, giving the variable's name, as shown in Figure 2-2.

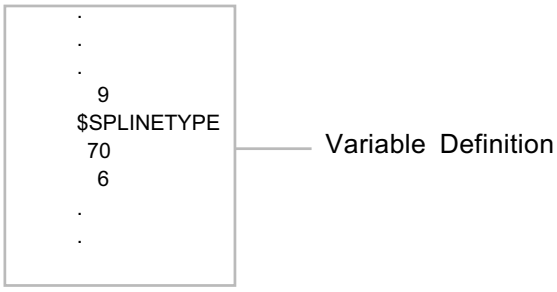


Figure 2-2 Example of Variable Definition

TABLES Section

The TABLES section of the DXF file contains AutoCAD tables associated with the drawing. The following tables may be included:

- APPID (not translated by **Cimatron**).
- DIMSTYLE (not translated by **Cimatron**).
- LTYPE - definition of line-types used
- LAYER - definition of layers used
- STYLE - fonts or shapes
- UCS - user's coordinate system
- VIEW (not translated by **Cimatron**).
- VPORT (not translated by **Cimatron**).

Table order may vary, but the LTYPE table must always precede the LAYER table.

Each table section begins with a table header, followed by a variable number of table entries, as shown in Figure 2-3. The end of each table is indicated by ENDTAB (end of table).

Each table entry begins with group code 0, identifying the table type, followed by group code 2, giving the name of the entry.

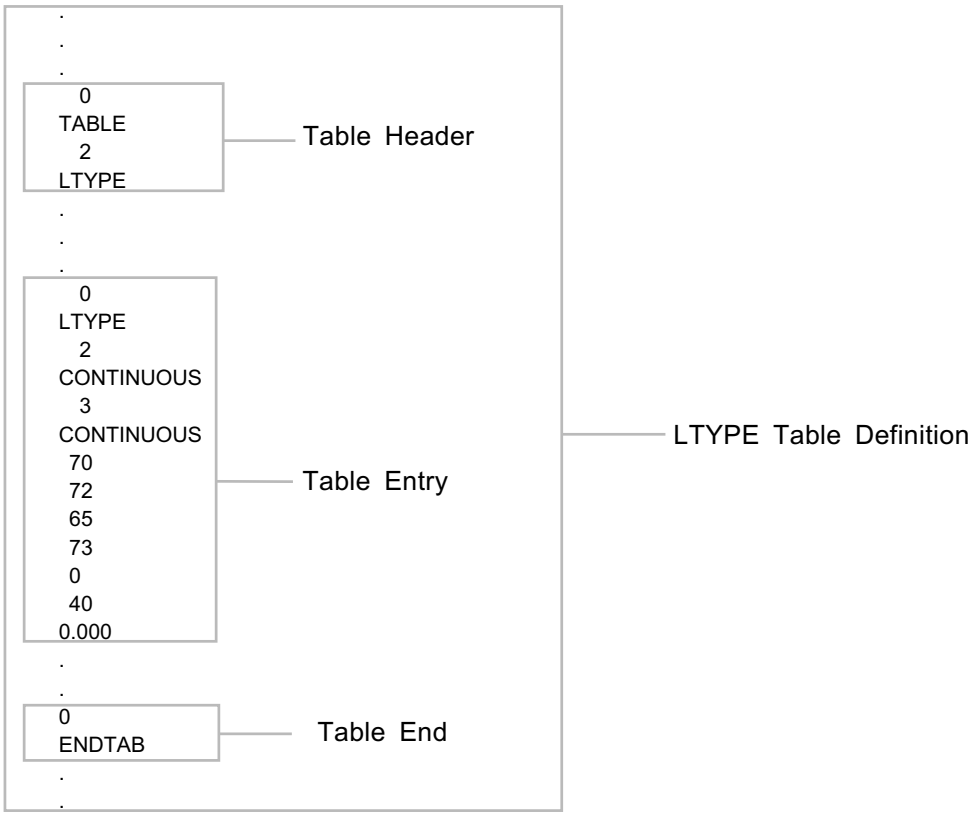


Figure 2-3: Example of Table Definition

BLOCKS Section

The BLOCKS section of the DXF file contains all AutoCAD BLOCK definitions associated with the drawing. Each BLOCK is introduced by a header, as shown in Figure 2-4. The header is followed by entities that make up the BLOCK. The end of each BLOCK is indicated by ENDBLK (end of block).

BLOCK definitions are never nested, i.e. no BLOCK and ENDBLK pair ever appears within another BLOCK and ENDBLK pair. Entities in BLOCK sections are identical to those in the entities section. Each entity begins with the group code 0.

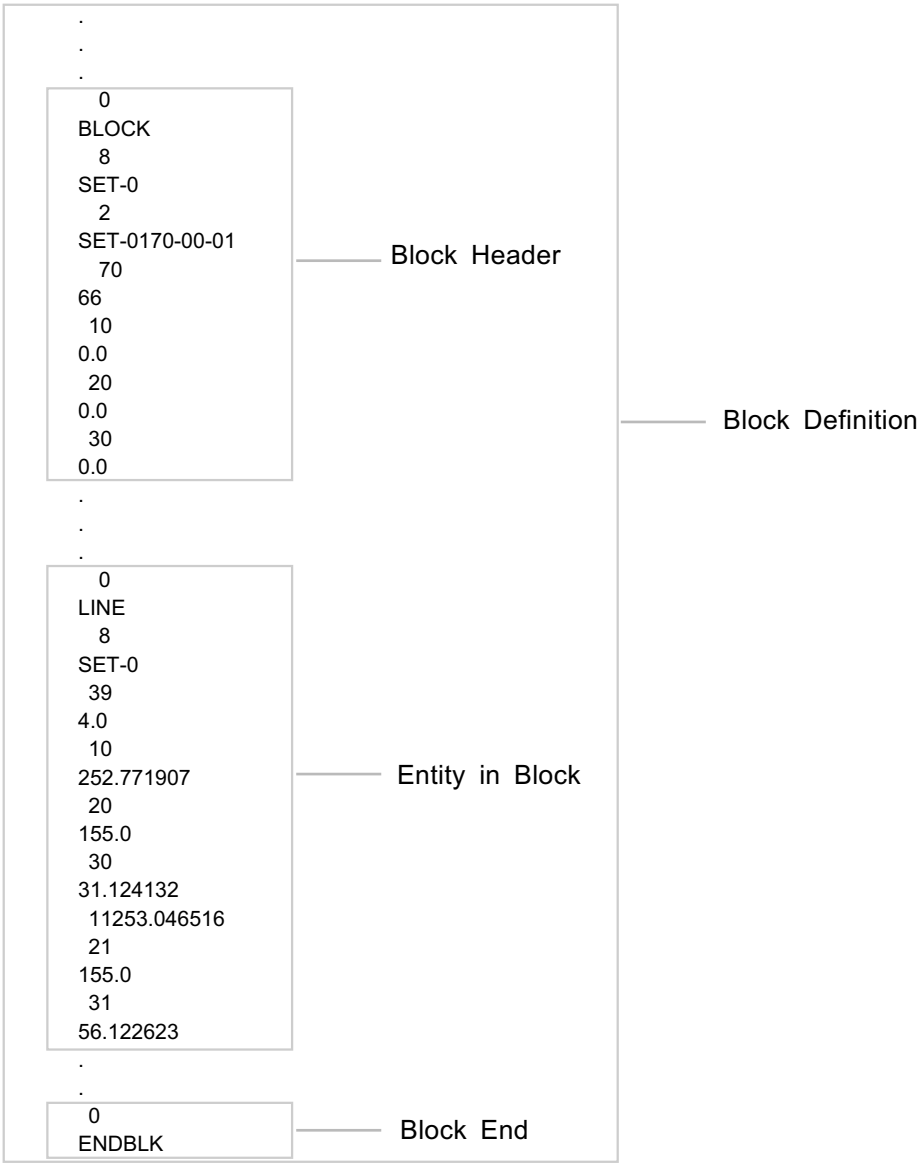


Figure 2-4: Example of BLOCK Definition

ENTITIES Section

The ENTITIES section of the DXF file contains all AutoCAD entities associated with the drawing. Each entity begins with the group code 0, as shown in Figure 2-5.

```
.
.
.
  0
LINE
  8
GEOM
 39
4.0
 10
252.771907
 20
155.0
 30
31.124132
 11
253.046516
 21
155.0
 31
56.122623
210
0.0000000000000001
220
0.9999999999999999
230
0.0
.
.
```

Entity Definition

Figure 2-5: Example of Entity Definition

❑

Chapter 3
The Read-DXF Utility

The **Read-DXF** Utility converts files in the DXF format to files in the **Cimatron** part file (**.pfm**) format.

Read-DXF uses a parameter file (**.dpr**). This file contains parameters that control the operation mode and determine the user interaction (refer to the section **Read-DXF Parameters**).

For more information about the parameter file and the format of its statements, refer to Chapter 5, **The DXF Utilities Parameter File**.

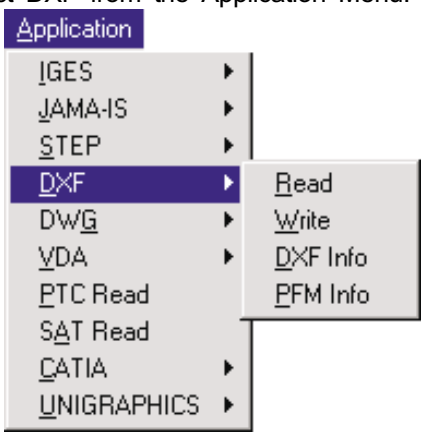
Translation data can either be displayed on the screen or optionally saved to a log file. The log file contains information such as user parameter values, a listing of each entity translated and any errors or problems encountered during its translation, and statistical information on the entities in both the DXF file and the **Cimatron** part file.

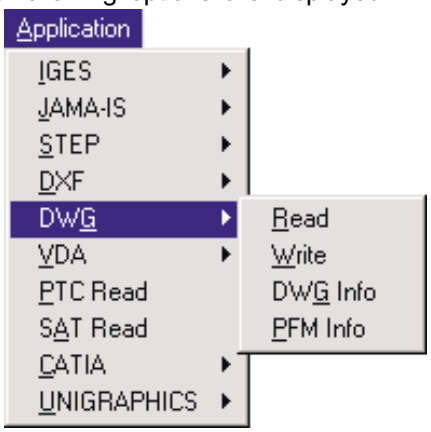
Running Read-DXF

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select DXF from the Application Menu. The following options are displayed.





3. Select the **DXF, Read** option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following DXF options are available:

```
1 -> Read option
2 -> Write option
3 -> Parameters File option
4 -> DXF File Information option
5 -> PFM File Information option
```

3. Select the **Read** option.

Running Modes

Read-DXF may be run interactively or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or: in **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

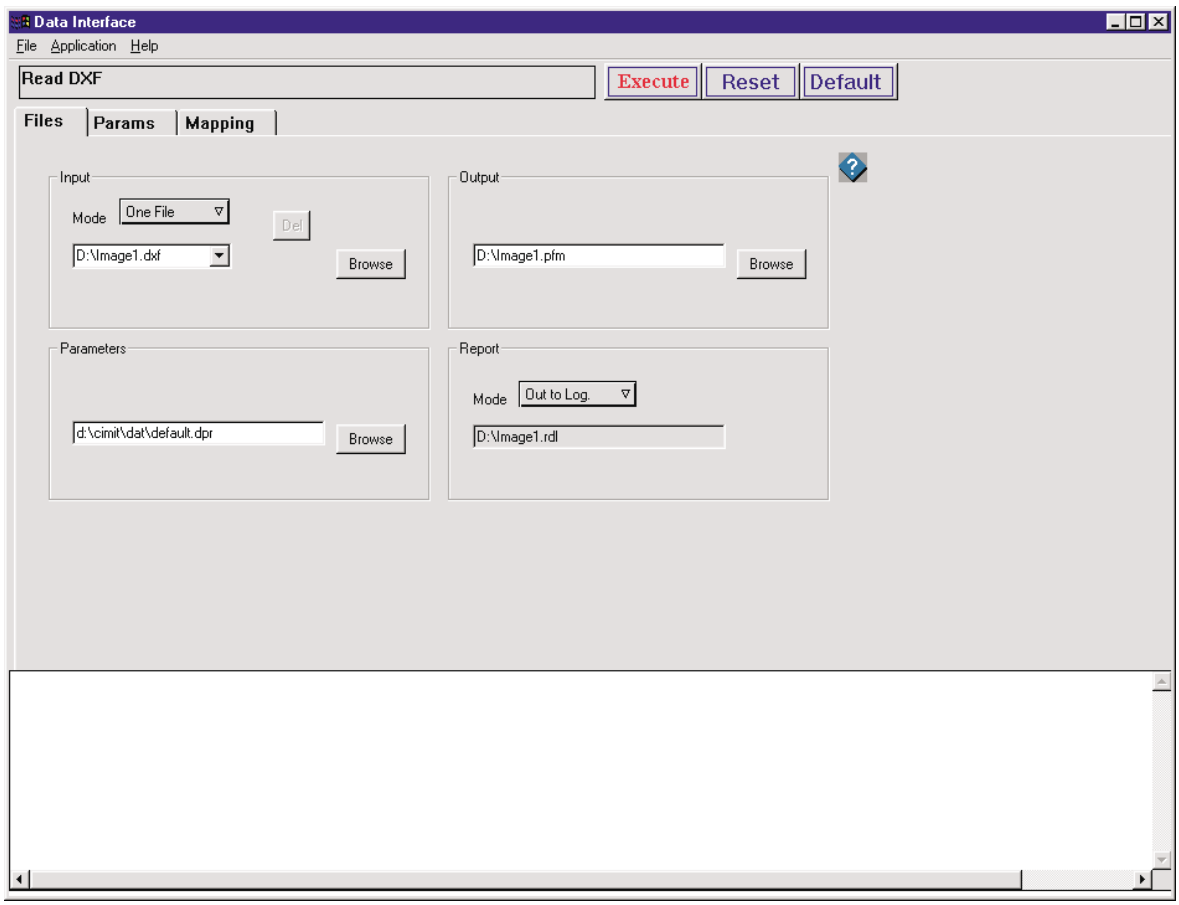
1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file containing one file name per line.

Notes:

- Multi mode is only available when you use the graphic interface for this utility.
- Each part file created will be in the same directory and have the same name as the DXF file (if no other name is entered), but will have the extension **.pfm** instead of **.dxf**.
- The DXF file may be located in the current or other directories. Enter the full path name of the DXF file if it is not located in the current directory.
- File extensions may be typed, unless otherwise specified. If the file name is not found as it is typed, the program will search for the same file name with the extension .dxf.
- In **Batch Mode Without Backup**, if a file already exists with the same name, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and **.pfm** files already exist with the name specified in the list, the old **.pfm** file will be copied to a backup file with the extension **.bac**. Then a new **.pfm** file will be created from the DXF file.

Graphic Interaction with Read-DXF



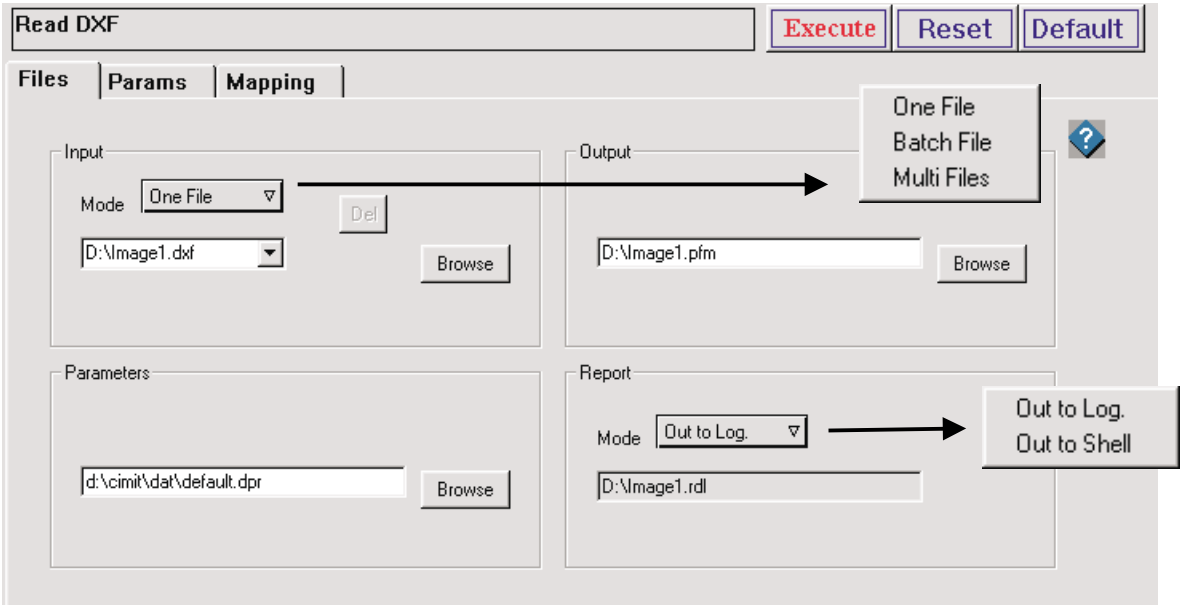
How To:

1. Select the appropriate tab for changing data where needed: **Files**, **Params** or **Mapping**.
2. Set the parameters.
3. Select a file in the Input area using the Browse button.
4. Press the Execute button.

The interaction for each of the tabs is described on the following pages.

Files

This dialog defines the Input, Ouput, Report and Parameter files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input **.dxf** file (the file to be converted to a Cimatron **.pfm**).

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.pfm** file is identical to the prefix of the input **.dxf** file.

   Choose a different **.pfm** file, or using the browse button, select an existing **.pfm** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the DXF parameter file is **default.dpr.** You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read DXF translation data **.rdl** file is identical to the prefix of the input **.dxf** file.

   Choose a different **.rdl** file, or using the browse button, select an existing **.rdl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

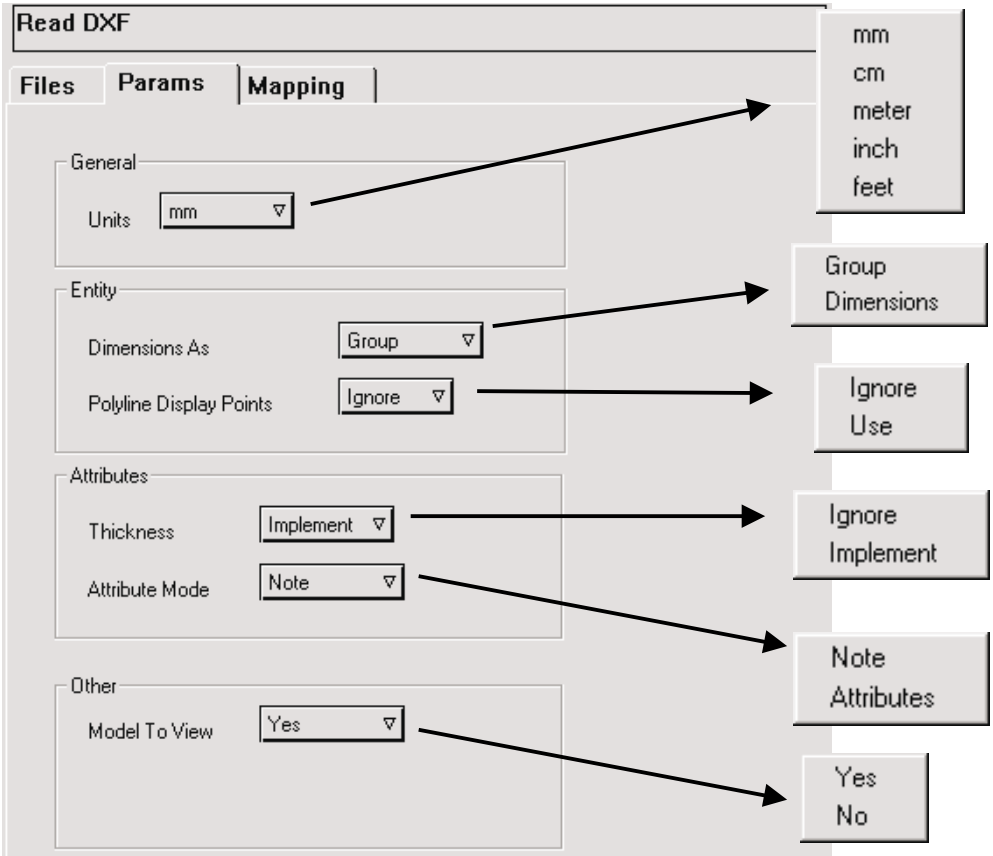
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

General.

1. Set the DXF file units of measurement with the equivalent Cimatron units.

Entity. (See Entity Mapping on page 3-13).

2. Read DXF dimensions as instances (Group) or as dimensions (Dimensions).
3. Use or ignore Polyline Display Points.

Attributes. (See Entity Mapping on page 3-13).

4. Implement or ignore thickness attributes during translation.
5. Read ATTDEF and ATTRIB entities from DXF either as Cimatron notes or as attributes.

Other.

6. Model To View:

   Yes = Translate all AutoCAD Model-Space entities to a dependant view.

   No = Translate all AutoCAD Model-Space entities to the Cimatron Model.

Mapping

This dialog defines the mapping characteristics for Text, Color, Pen and Line Font.

How To:

Text.

1. Map the source (DXF) font with the equivalent destination (Cimatron) font.

Color.

2. Convert a DXF color to an equivalent Cimatron color.

Pen.

3. Convert a DXF color to a Cimatron pen.

Line-Font

4. Convert a DXF line font to an equivalent Cimatron line font.

Text Conversion

Maps the source (DXF) font with the equivalent destination (Cimatron) font.

Characters from the source font are mapped to destination font characters using the character mapping file (**.cdt** file) and the char-map-area-number.

Character mapping files are located in the directory **\dat\fonts**. The char-map-area-number is a hexadecimal number indicating the range of source font character codes to be used from the character mapping file when the given source font characters are mapped. Each char-map-area-number specifies a range of 256 characters.

The **.cdt** files used by DXF Utilities are:

rdxf.cdt	for the Read-DXF option
dxf-font	DXF Font Code number of the DXF Font
cim-font	Cimatron Font name, one of the fonts located in the directory **\dat\fonts**.
char-map-area-number	Area number (Hexadecimal number) in the character mapping file (**.cdt** file).

Defaults: If the MAP TEXT statement for a specific font is omitted, the defaults are:

Default Cimatron font for Read-DXF	standard
Default char-map -area-number if omitted	0

Color Conversion

Convert a DXF color to the equivalent Cimatron color.

Default: As defined in the default parameters file.

Note:

- The defaults are set for AutoCAD predefined colors. Other colors are converted to Cimatron color 1.

Pen Conversion

Convert a DXF color to a Cimatron pen.

Defaults: a DXF color will be converted to Cimatron pen 1.

Line-Font Conversion

Convert the DXF line-type to the equivalent Cimatron line-font.

Cimatron Line Font #	Type of Line Font
1	Solid
2	Dash
3	Center
4	L-Dash
5	Font - 5

Default: as defined in the default parameter file.

Notes:

- The defaults are set for AutoCAD standard line-types.
- Other line-types are mapped with Cimatron line-font 1 (SOLID).

Textual Interaction with Read-DXF

The following section describes the interaction upon invoking **Read-DXF**. (The interaction that is shown will occur when all the interaction parameters in the parameter file (**.dpr**) are set to ASK.) All prompts that appear on the screen are written here in italics.

DXF (version 2000.0) - Cimatron (version 11.0)
==========

Enter DXF parameters file name (<CR> = default):	Type name of a parameter file; do not type the **.dpr** extension. To use the default parameter file (**default.dpr**), press <CR>.
Available modes:	The mode menu will appear.
1 -> Interactive mode *2 -> Batch mode with backup* *3 -> Batch mode without backup*	
Enter mode (0 = exit)	Type the number of the appropriate operation mode. To exit the program at this point, type 0 (zero).

If Interactive mode is selected:

Enter DXF file name (<CR> = mode menu)	Type the name of the DXF file to be converted; do not type the **.dxf extension.** To exit the program at this point, press <CR>.
Enter part file name (<CR> = <file_name>) :	Type a name for the part file. To use the name displayed in parentheses, press <CR>. If the specified file already exists, the following message appears:

—- File <file_name>.pfm already exists

Replace existing file ? (Y/N)	To write over the existing file, type **Y**. To type a different name for the file, type **N**.
Save DXF log file (Y/N) ? (<CR> = Y) :	You will be asked whether you want to save the **Read-DXF** log file. If you enter **y**, all translation data will be saved to a log file. The name of the log file will be that of the DXF file with the extension **.rdl**. If you enter **n**, the translation data will be displayed on the screen. Reading DXF file ...
Setting up part file <file_name> ...	The new part file will be saved with the **.pfm** extension.

Attribute Mapping

The DXF attributes, a set of group codes common to all entities, are mapped to **Cimatron** attributes as follows:

- Entity Handle (group code 5)
  A unique, permanent entity identifier. This attribute is ignored.
- Layer (group code 8)
  The name of the layer on which the entity resides. Layer is mapped with the **Cimatron** Level attribute as follows.
  - When no layer is specified, **Cimatron** 'LEVEL0' is assigned.
  - When the layer name exceeds 6 characters, the **Cimatron** Level will be assigned an abbreviated name, containing the first five characters of the DXF layer name, followed by a sequential number.

    For example, mapping the following DXF layers gives these results:

    TOP-VIEW→TOP-V1
    SIDE-VIEW→SIDE-2
- Color (group code 62)
  Color is mapped with the **Cimatron** Color attribute through the parameters file. Color may also be mapped with Cimatron Pen Attribute through the parameters file.
- Linetype (group code 6)
  Linetype is mapped with **Cimatron** Line-Font attribute through the parameters file.
- Extrusion Vector (group codes 210, 220, 230)
  A 3D vector describing the Z axis of the ECS (Entity-associated Coordinate System). It is mapped with the **Cimatron** rotation pointer using the arbitrary axes algorithm (refer to the AutoCAD Customization manual).
- Elevation (group code 38)
  The Z distance above or below the XY plane of the ECS. Exists only in DXF versions prior to 11.0.
- Thickness (group code 39)
  Extrusion amount from ECS in positive direction of the ECS Z axis (refer to Table 2-1).
- Space Flag (group code 67)
  When the DXF space flag = 1 (paper space), the entity will be created in the Drafting application, otherwise it will be created in the Modeling application.

Entity Mapping

Table 3-1: DXF Entities Supported by the Read-DXF Utility

DXF Entity		Cimatron Entity			Notes
Name	Thickness (Group Code 39)	Name	Class	Type	
POINT	No Yes	Point 2 Points & connecting Line	0 0 1	1 1 1	(1)
LINE	No Yes	Line Drive Surface	1 3	1 2	
CIRCLE	No Yes	Arc Drive Surface	1 3	2 2	
ELLIPSE		Conic	1	3	
SPLINE		2D Spline 3D Spline	1 2	4 4	
ARC	No Yes	Arc Drive Surface	1 3	2 2	
TRACE	No Yes	Planar Face 2 Planar Faces & 4 Drive Surfaces	4 4 3	1 1 2	
SOLID	No Yes	Planar Face 2 Planar Faces & 4 Drive Surfaces	4 4 3	1 1 2	(2)
3DFACE		Planar Face when 3 edges or 4 co-planar edges; Ruled Surface otherwise	4 3	1 3	(7)
POLYLINE		Curves, Surfaces			(3)
VERTEX					Translated as part of POLYLINE
SEQEND					Translated as part of POLYLINE or INSERT
BLOCK		Internal master	14	11	(4)
ENDBLK					Translated as part of BLOCK.
INSERT		Instance	13	1	(5)
TEXT		Note	6	5	(6)
DIMENSION		Instance or Dimension	13 5	1	(8)
MTEXT		Note	6	5	
3Dsolid		Unsupported entity			
Attribute Definition		Unsupported entity			
Body		Unsupported entity			
Leader		Instance	13	1	
Ray		Unsupported entity			
Region		Unsupported entity			
Shape		Unsupported entity			
Viewport		Instance	13	1	
Xline		Unsupported entity			
Proxy		Instance	13	1	

Notes:

- All DXF POINT shapes are mapped into one **Cimatron** Point shape. POINT orientation does not exist in **Cimatron**.
- If only three edges exist, three Drive surfaces and two Planar Faces will be created.
- DXF Curves and surfaces are defined under the Polyline entity. Table 3-2 shows the mapping of Polyline entities to **Cimatron** entities:

Table 3-2: Mapping of DXF Polyline Entities to Cimatron Entities

DXF Entity			Cimatron Entity		
Polyline Type (Group Code 70)	**Smooth Type (Group Code 75)**	**Thickness (Group Code 39)**	**Name**	**Class**	**Type**
2D Polyline		No	3D Comcrv	2	15
	5	No	2D Spline(NURBS)	1	4
	6	No	2D Spline(NURBS)	1	4
		Yes	Drive Surface	3	2
	5	Yes	Drive Surface	3	2
	6	Yes	Drive Surface	3	2
3D Polyline			3D Comcrv	2	15
	5		3D Spline(NURBS)	1	4
	6		3D Spline(NURBS)	1	4
3D Polygon Mesh			Bezier Surface	3	12
	5		B-spline Surface	3	11
	6		B-spline Surface	3	11
	8		B-spline Surface	3	11
Polyface mesh			Unsupported entity		
Linetype pattern to be generated continuously through the vertices			Translated with default pen		

If a Composite Curve is composed of one entity, the Composite Curve primitive will be created.

DXF variable width polylines are translated to lines of uniform width in **Cimatron**.

Polyline Smooth type values are:

- 5 quadratic B-spline
- 6 cubic B-spline
- 8 Bezier

Refer to the parameter R-POLYLINE-DISPLAY-POINT in Chapter 5, **The DXF Utilities Parameter File.**

- DXF external reference BLOCKS are translated to external Masters.

  Each DXF file is converted to a PFM file. The PFM file will be placed at the same location as the DXF file.

- If different axis factors were used on a DXF BLOCK, the X axis factor will be used on all axes when scaling in **Cimatron.**

  An inserted BLOCK cannot have thickness in **Cimatron.**

- More than one note may be created, since text may be mapped with different fonts.

  The following are not implemented in **Cimatron:**

  - Oblique angles
  - Mirrored text
  - Text Thickness
  - Text width

  Text aligned between two points in a text-box will not remain within the boundaries of the box in **Cimatron**.

- Edge visibility is not implemented in Cimatron.

- DXF dimension entity supported by the Read-DXF Utility. If a dimension is read as a "group" from a parameter file (**.dpr**), **Cimatron** translates the group as an Instance. If the dimension is read as a "dimension", **Cimatron** translates it as a dimension.

  Table 3-3 shows the mapping of a DXF dimension to a **Cimatron** entity:

Table 3-3: Mapping of DXF Dimension to Cimatron Entity

DXF Entity	Cimatron Entity		
Dimension entity (Group Code 70)	Name	Class	Type
Rotated, horizontal, or vertical	Linear	5	1
Aligned	Linear	5	1
Angular	Angular	5	3
Diameter	Radial	5	2
Radius	Radial	5	2
Angular 3-point	Angular	5	3
Ordinate	Ordinate	5	4

❑

Chapter 4
The Write-DXF Utility

Write-DXF converts **Cimatron** part files (**.pfm**) to files in the DXF format.

Write-DXF uses a parameter file (**.dpr**). This file contains parameters that control the operation mode and determine the user interaction as well as optional user filters which provide selective translation of DXF entities. For more information about the parameter file and the format of its statements, refer to Chapter 5, **The DXF Utilities Parameter File**.

Translation data can either be displayed on the screen or optionally saved to a log file. The log file contains information such as user parameter values, a listing of each entity translated and any errors or problems encountered during translation, and statistical information on the entities contained in the **Cimatron file.**

Running Write-DXF

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select DXF from the Application Menu. The following options are displayed.
3. Select the **DXF, Write option.**

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following DXF options are available:

```
1 -> Read option
2 -> Write option
3 -> Parameters File option
4 -> DXF File Information option
5 -> PFM File Information option
```

3. Select the **Write** option.

Running Modes

Write-DXF may be run interactively or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or: in **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

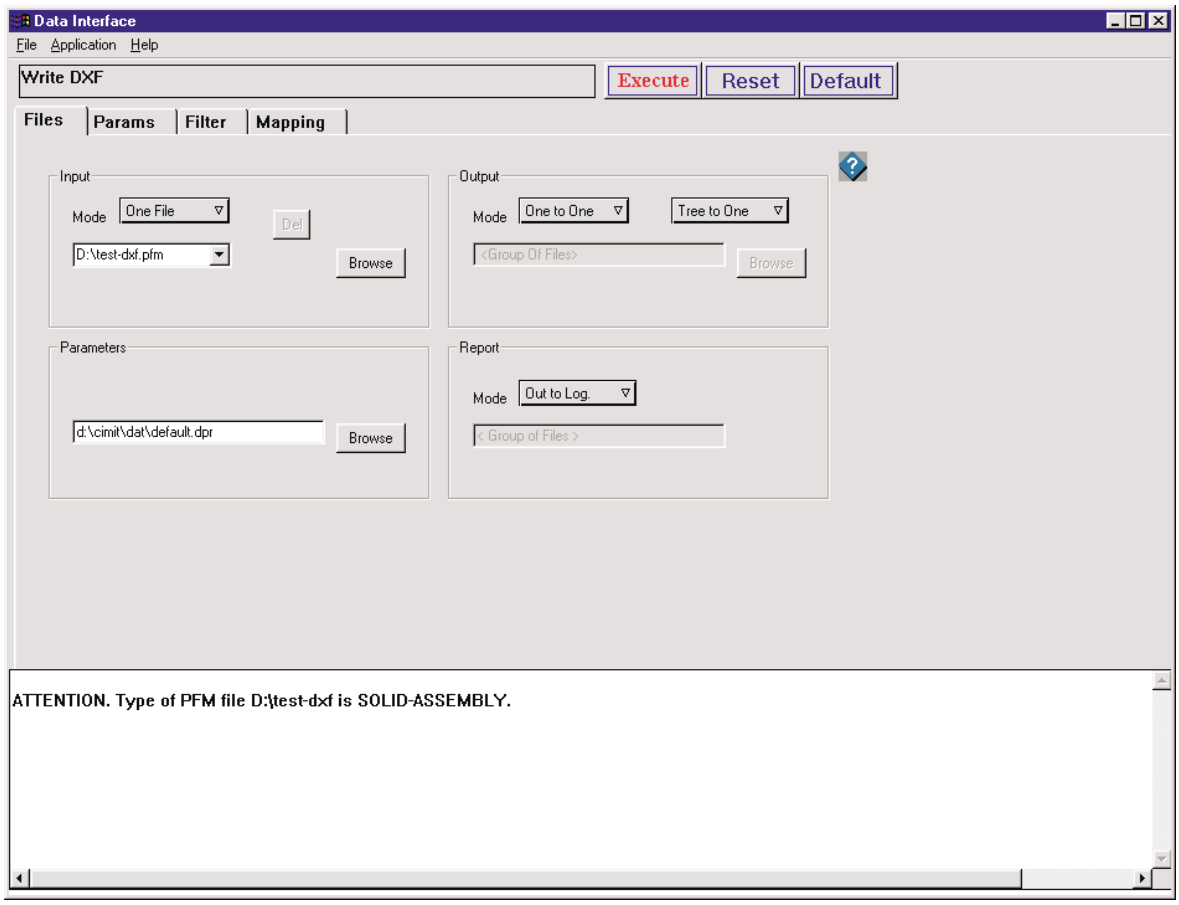
1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file containing one file name per line.

Notes:

- Multi mode is only available when you use the graphic interface for this utility.
- Each DXF file created will be in the same directory and have the same name as the **Cimatron** part file (if no other name is entered), but will have the extension **.dxf** instead of **.pfm**.
- The part file may be located in the current or other directories. Enter the full path name of the part file if it is not located in the current directory.
- File extensions may be typed, unless otherwise specified. If the file name is not found as it is typed, the program will search for the same file name with the extension **.dxf**.
- In **Batch Mode Without Backup**, if a file already exists with the same name, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and **.dxf** files already exist with the name specified in the list, the old **.dxf** file will be copied to a backup file with the extension **.bac**. Then a new **.dxf** file will be created from the **Cimatron** part file.

Graphic Interaction with Write-DXF



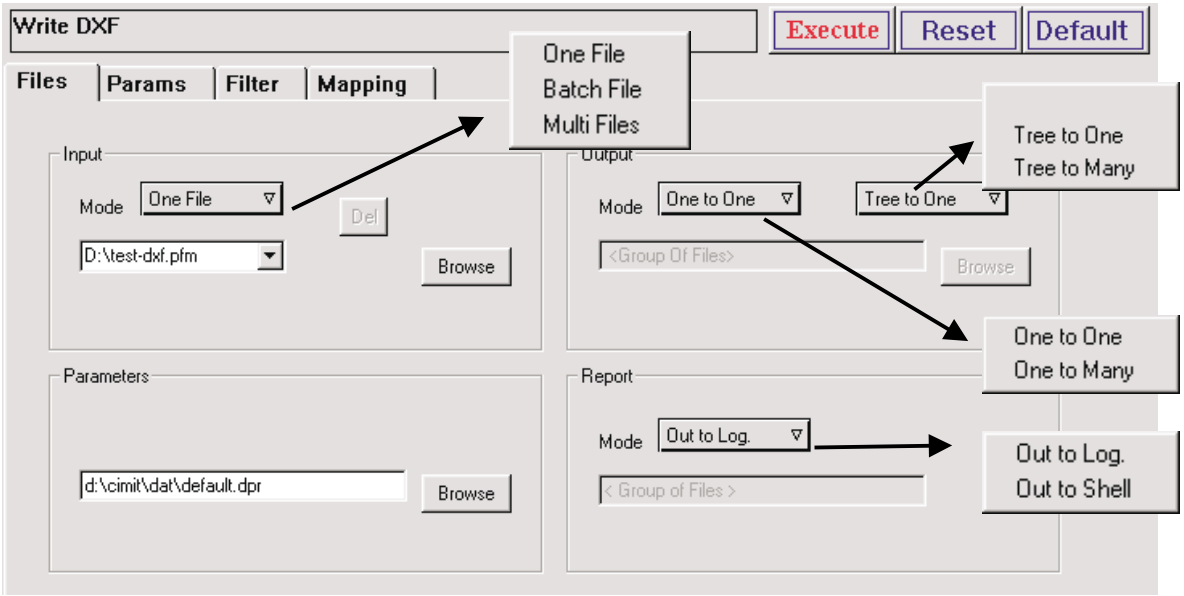
How To:

1. Select the appropriate tab: **Files**, **Params, Filter** or **Mapping**.
2. Set the parameters.
3. Select a file in the Input area using the Browse button.
4. Press the Execute button.

The interaction for each of the tabs is described on the following pages.

Files

This dialog defines the Input, Ouput, Report and Parameter files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Mode: One File, Batch File or Multi Files.
2. Select the input **.pfm** file (the file to be converted to a DXF file).

The available options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.dxf** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.dxf** file, or using the browse button, select an existing **.dxf** file. In the latter case, you will be prompted whether or not to overwrite the existing file.
2. Select an Application Filter: ONE TO ONE, ONE TO MANY or ONE TO TWO.
3. Select an Assembly Filter: TREE TO ONE or TREE TO MANY.

The following Application Filters define if an entity is to be translated:

- ONE TO ONE

  All available arguments (all defined views, drawings and model) are included in one file.

  Entities are included according to their visibility in the application. For example, an entity that is visible in an included view will be translated, even if the model in which the entity was created is not included. To give another example, including a drawing causes all views referenced within it to be written to the file.

- ONE TO MANY

  Split the PFM file to many output files. The LOG FILE is not available in this option because many log files are created.

  This option enables you to write the model, each view, or each drawing of a PFM file to a different file.

  The output files are placed in a directory named after the PFM file.

  The view/drawing file name prefix is identical to the translated view/drawing name with an addition (_d for drawing, _v for views).

  The model file name prefix is model.

  If the name of the view or drawing contains special characters, such as a dot (.), dollar sign ($), etc., they will be converted to an underscore (_) in the output file name.

The following Assembly Filters are available:

- TREE TO ONE — Create *one* DXF file for the assembly model and one DXF file for the drafting model.

  DXF model file name:

  assem_mod.dxf.

  DXF drafting file name:

  assem_draft.dxf.

- TREE TO MANY — Create a directory with *many* DXF files for the assembly model; one DXF file for each part file in the assembly and one DXF file for the drafting model.

  DXF model file names:

  part1.dxf, part2.dxf, etc.

  DXF drafting file name:

  assem_drf.dxf

Parameters

How To:

1. By default, the prefix of the DXF parameter **.dpr** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.dpr** file, or using the browse button, select an existing **.dpr** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

The **parameter file** enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Write DXF translation data **.wdl** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.wdl** file, or using the browse button, select an existing **.wdl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

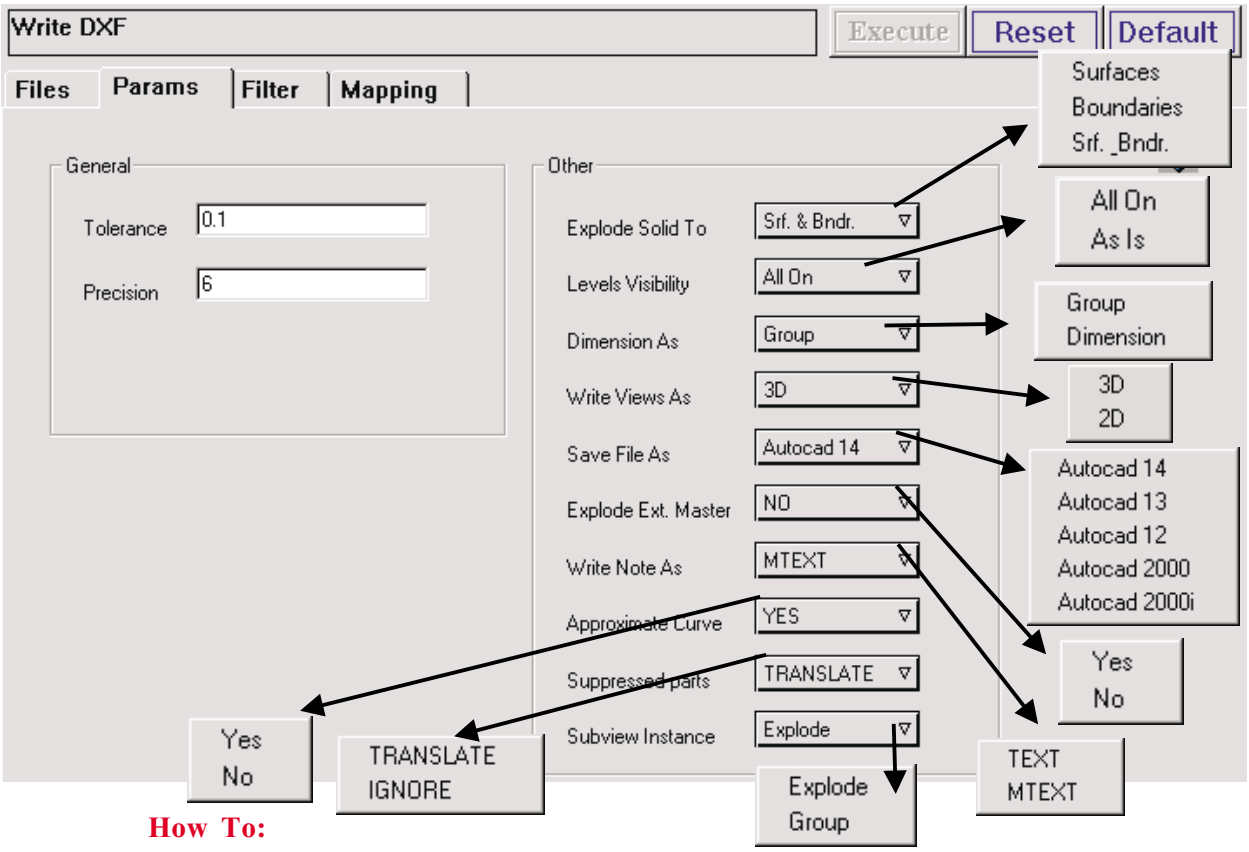
The available options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

General.

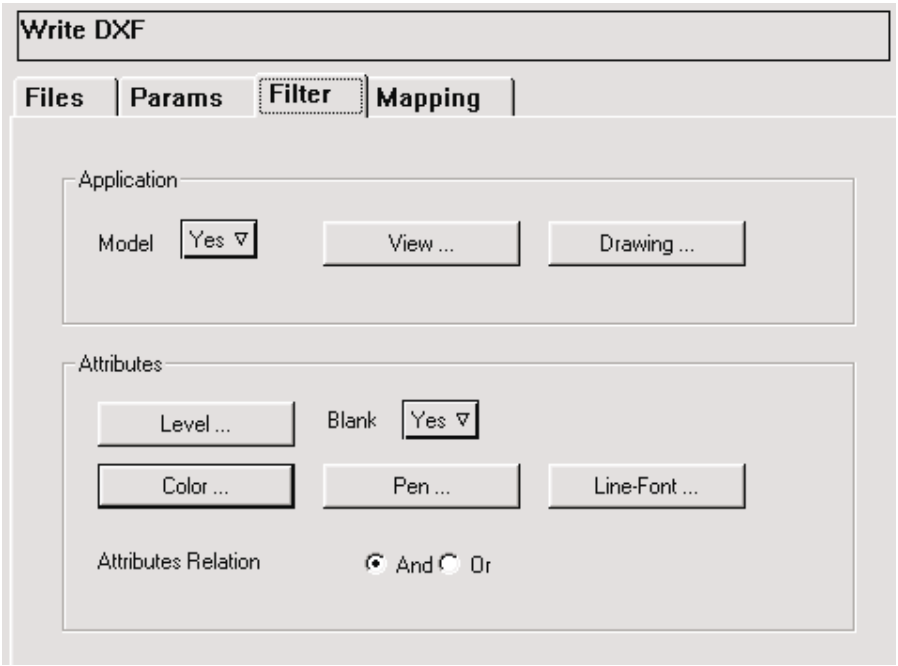
1. Set the tolerance in MM when approximating a Cimatron entity by an entity in another standard. The tolerance range is 0.001 - 100 (Default = 0.1). If the text in the window is deleted, the last entry is selected.
2. Set a Precision digit (range 1-9, default = 6).

Other.

3. Explode a Solid model to Surfaces, Boundaries or Surfaces & Boundaries. (Default = Surfaces & Boundaries)
4. Set the Levels Visibility as in the Cimatron part file or set all levels on.
5. Write dimensions as instances (Group) or as dimensions (Dimensions).
6. Write views as 2D or 3D.
7. Save the file as an AutoCAD version 12, 13, 14, 2000 or 2000i file.
8. Choose whether or not to explode external masters.
9. Write notes as AutoCAD TEXT or AutoCAD MTEXT (TEXT is a one line text entity, and MTEXT is a multi-line text entity).
10. Select how to approximate Cimatron curves; either YES to approximate all Cimatron curves to polygons (default), or NO to translate Cimatron ellipse entities to AutoCAD ellipses and Cimatron spline entities (2D or 3D) to AutoCAD splines .
11. Choose whether to ignore or translate suppressed parts.
12. Choose how to translate Subview Instances in drawings - either to Group them and translate them as inserts (default), or to Explode them and translate them as entities belonging to the drawing.

Filter

The dialog box below defines the filters used during the file translation.



How To:

Application

1. Include (Yes) or exclude (No) the model in the translation. An entity that is visible in an included view will be translated.
2. Select views to be translated. All views in the **Selected Views** column will be included in the translation. If more than 47 views have been selected, only the first 47 will be translated.
3. Select drawings to be translated. All drawings in the **Selected Drawings** column will be included in the translation. If more than 47 drawings have been selected, only the first 47 will be translated.

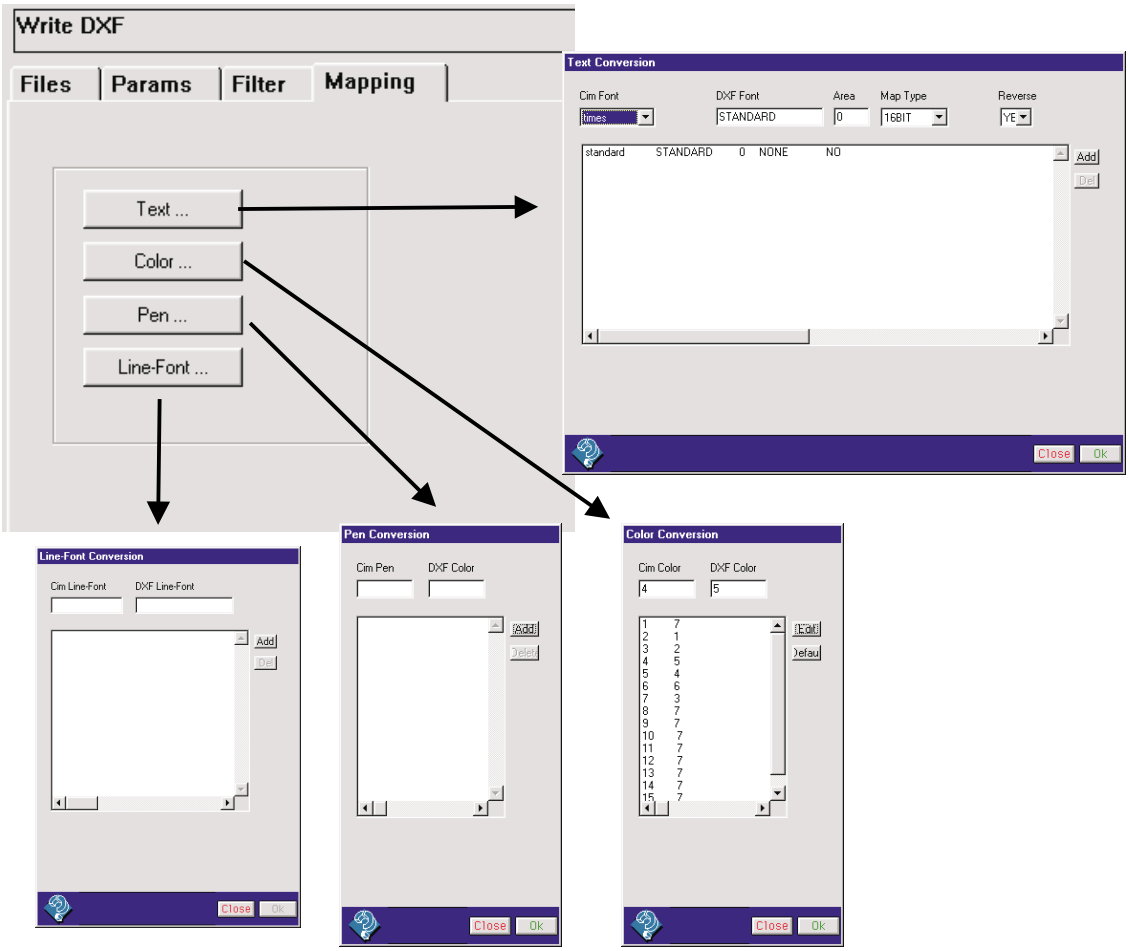
Attributes

Attribute filters allow entities with specified attributes to be included.

4. Select levels to be translated. All levels in the **Selected Levels** column will be included in the translation.
5. Include (Yes) or exclude (No) blanked entities in the translation. The Blank filter is useful when blanked entities are not needed in the output file.
6. Specify a color to cause entities with this color to be included.
7. Specify a pen to cause entities with this pen type to be included.
8. Specify a line font to cause entities with this line font to be included.

Mapping

This dialog defines the mapping characteristics for Text, Color, Pen and Line Font.



How To:

Text.

1. Map the source (Cimatron) font with the equivalent destination (DXF) font.

Color.

2. Convert a Cimatron color to an equivalent DXF color.

Pen.

3. Convert a Cimatron pen to a DXF color.

Line-Font

4. Convert a Cimatron line font to an equivalent DXF line font.

Textual Interaction with Write-DXF

The following section describes the interaction upon invoking **Write-DXF**. (The interaction that is shown will occur when all interaction parameters in the parameter file (**.dpr**) are set to ASK.) All screen displays are shown here in italics.

Cimatron (version 11.0) DXF (version 2000.0) - Convertor <version number>
==========

Enter DXF parameters file name (<CR> = default):	Type the name of the parameter file without the **.dpr** extension. To use the default parameter file, press <CR>.
Available modes:	The mode menu will appear.
1 -> Interactive mode *2 -> Batch mode*	
Enter mode (0 = exit)	Type the number of the appropriate operation mode. To exit the program at this point, type 0 (zero).

If Interactive mode is selected:

Enter part file name (<CR> = mode menu) :	Type the name of the part file without the **.pfm** extension. To return to the mode menu, press <CR>.
Enter DXF file name (<CR> = <file_name>) :	Enter a name for the DXF file to be created. To use the name displayed in parentheses, press <CR>. If the specified file already exists, the following appears:
—File<file_name>.dxf already exists *Replace existing file? (Y/N)*	To write over the existing file, type **Y**. To enter a different name for the new file, type **N**.
Save DXF log file (Y/N) ? (<CR> = Y) :	You will be asked whether you want to save the **Write-DXF** log file. If you enter **Y**, all translation data will be saved to a log file. The name of the log file will be that of the DXF file with the extension **.wdl**. If you enter **N**, the translation data will be displayed on the screen.
Setting up part file <file_name> . . .	

Application Mapping

Cimatron implements an architecture of model-views-drawings with multiple views and/or drawings. Inside each view application a different projection of the model can be made, model entities can be deleted or trimmed and view-specific entities can be created. A drawing consists of one or more views together with drawing specific entities.

The DXF file structure implements an architecture of model-views-drawing with multiple views, but only one drawing. Each view can include a portion of the model without any modifications to model entities, or additions of entities that are view-specific. The drawing consists of views and drawing specific entities. This DXF file architecture is available from AutoCAD version 11. Up to version 11, only one drawing was possible, including the model and the drawing data.

Because of this difference in architecture, and in order to maintain compatibility with previous DXF versions, the **Write-DXF** utility writes all **Cimatron** data to the DXF default application, which is the model.

When more than one application exists in **Cimatron** and **Write-DXF** is used, applications can overlap each other in the generated DXF file. To overcome this problem, the **Write-DXF** utility provides the user with user-defined filters through the DXF utilities parameter file. The user can define which application to write, and only this application data will be written to the DXF file. For a detailed description, refer to the section **Selective Translation of Cimatron Files to DXF Files**.

Attribute Mapping

Cimatron attributes are mapped to DXF attributes, a set of group codes that are common to all entities, as follows:

Level	Translated as DXF Layer attribute (group code 8).
Color	Mapped with the DXF Color attribute (group code 62) through the parameters file. If a **Cimatron** color can be mapped with a pre-defined AutoCAD color, this will override any mapping in the parameters file.
Line-font	Translated to the DXF Linetype attribute (group code 6) preserving the name and pattern of the **Cimatron** line-font.
Pen	No equivalent attribute in DXF attributes. May be mapped to DXF Color attributes through the parameters file. In the event of a conflict between color mapping and pen mapping, pen mapping overrides the color mapping.
Blank Status	No equivalent attribute in DXF attributes.
Rotation Pointer	Mapped with the DXF extrusion vector (group codes 210, 220, 230) describing the Entity Associated Coordinates System (ECS).

Entity Mapping

Table 4-1: Entities Supported by the Write-DXF Utility

Cimatron Entity			DXF Entity Name	Note
Name	**Class**	**Type**		
Point	0	1	Point	
Line	1	1	Line	
Arc	1	2	Arc, Circle	
Other 2D Curves	1		Polyline	
3D Curves	2		Polyline	
Surfaces	3		Polyline	
Planar Face	4	1	Set of its boundaries	
Linear Dimension	5	1	Block and Insert	(1)
Radial Dimension	5	2	Block and Insert	(1)
Angular Dimension	5	3	Block and Insert	(1)
Ordinate Dimension	5	4	Block and Insert	(1)
Chamfer Dimension	5	5	Block and Insert	(1)
Arc-Len Dimension	5	6	Block	
Geometric Tolerance	5	10	Block and Insert	(1)
Section Line	6	1	Block and Insert	(1)
Hatch	6	2	Block and Insert	(1)
Note	6	5	Text	
ID Num	6	7	Block and Insert	(1)
B.O.M.	6	9	Block and Insert	(1)
Label	6	10	Block and Insert	(1)
Instance	13	1	Insert	
Internal Master	14	1	Block	
Group	14	9	Block and Insert	(1)
Sub-view Reference	14	3	Block and Insert	

Note:

- When a **Cimatron** entity is translated as a DXF Block and an Insert, the block is named by an identifying letter followed by the block number. The following table gives the correspondence between entity type and identifying letter:

Cimatron Entity Type	Identifying Letter
Dimensions	D
Geometrical Tolerance	T
Section Line	S
Hatch	H
Note	N
ID-Num	I
Label	L
Group	G
B.O.M.	B

For example, a **Cimatron** Label entity with ID = 50 is translated as a Block named “*L50" and an Insert.

Constraints

- Slave entities are never written by themselves, but as part of their parent entity, i.e. only independent parent entities can be filtered or unfiltered.
- Master entities are always written. As a result, a level that is excluded by the user may be written to the DXF file because of Master entities that are located on the level.
- Instances and Groups are written according to application group filters only. Their inclusion is not affected by their attributes.
- There are limits on the number of argument values accepted for some attribute filters, as shown in Table 4-3.

Table 4-3. Maximum Number of Argument Values for Attribute Filter Arguments

Filter Argument	Maximum No. of Values
DEPENDENT VIEW	47
LEVEL	500
COLOR	15
LINE-FONT	8
PEN	8
SET	1

- When a view is written to DXF, the view point of this is written as VPORT in the DXF. This means that the view written to DXF will be displayed in Autocad exactly as displayed in **Cimatron**.

❑

Chapter 5
The DXF Utilities Parameter File

The **DXF** Utilities parameter file allows the **DXF** utilities user to:

- customize the interaction with the utilities.
- insert data required by the utilities.

This chapter contains a general description of the creation, format and use of the parameter file.

A description of the **Parameter File Check** Utility is given at the end of this chapter.

Parameter File Name

The format of the parameter file name is **filename.dpr**. There are no limitations on the name, however the extension must be **.dpr**. For example, the name of the default parameter file is **default.dpr**.

Parameter File Creation

The **Cimatron** DXF parameter file is created by the user. Since the file is in ASCII (American Standard Code for Information Interchange) format, it can be easily created using any text editor. The Cimatron original default parameter file (**default.dpr**), located in the **<root_cad>\dat** directory, can be used as a template and guide for the user.

You can create the parameter file by selecting **File >> Save Params As...**.

Parameter File Format and Rules

The parameter file is an ASCII file. The file contains statements which consist of predefined keywords and their parameter values. Following are some general rules of the parameter file format:

- The order of the lines is not important.
- Letter case is not important except when specifying **Cimatron** files, in which case the **Cimatron** file naming rules must be used.
- Each line starts with a keyword.
- If the same keyword appears more than once, and the statements can be accumulated, they will; otherwise the last statement encountered will be used.
- A line that starts with ## is a comment line.
- Empty lines are ignored.
- Maximum of 50 words are allowed in one statement.
- To continue the arguments of a statement onto an additional line, finish the current line with \<CR> (backslash followed by carriage return) and continue on to the next line.
- The maximum number of characters allowed in one statement is 256.

Parameter File Statements and Syntax

All the statements in the parameter file have the following format:

```
KEYWORD1 KEYWORD2 statement-argument(s) statement-switch(es)
```

In general, statement arguments are required while statement switches are optional. Statements are divided into three groups according to their function. The first keyword of each statement identifies which group it belongs to:

- SELECT statements are used to select one of a number of predefined options. SELECT statements usually require only one argument. Additional arguments or switches are optional. If an optional argument is not supplied by the user, a default program value is used.
- SET statements provide specific information that is required by a utility.
- MAP statements are used to map values between two sets of data.

Following are the conventions used to describe the parameter file statements:

- | - separates different options of the same argument.
- {A} - argument is required.
- [A] - argument is optional.
- <A> - argument is the default value if no value is given.
- (n-m) - range of possible values.

- Upper case words are fixed predefined keywords or arguments values. They should be typed in the parameter file exactly as they appear here.
- Lower case words are used for variables for which the user would normally substitute a value.
- The keyword ASK means that the user will be prompted to supply information when running the program.

Read-DXF Parameters

The parameter file (**.dpr**) contains parameters that control the operation mode, determine the interaction when running **Read-DXF** and provide information needed by the utility. These parameters will be described here.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP |<NOBACKUP>]

  The parameter R-MODE selects the mode of operation.
  Read-DXF may be run interactively or in batch mode. Optional backup is available with batch mode.

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. User is prompted for name of each DXF file to be converted.
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the DXF files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.
BACKUP	Option used in batch mode. If a **.pfm** file already exists with the name specified in the batch file, it will be copied to a backup file with the extension **.bac**.
NOBACKUP	This is a system default option used in batch mode. If a part file (**.pfm**) with the name specified in the list already exists the new part file will replace it.

  Defaults: ASK, NOBACKUP

- SELECT R-LOG-FILE {YES | NO | <ASK>}

The parameter R-LOG-FILE selects log file mode.
Translation process data can either be saved to a log file or displayed on the screen.

YES	All translation data will be saved to a log file. The name of the log file will be that of the DXF file with the extension **.rdl**. The screen will display only information about the DXF type and form contained in the DXF file.
NO	Translation data will be displayed on the screen.
ASK	User is asked whether he wants to save translation data to log file.

Default: ASK

Statements for Conversion

- SELECT R-THICKNESS {<IMPLEMENT> | IGNORE}

Selects the thickness attribute implementation flag (refer to the section **Entity Mapping)**.

IMPLEMENT	Thickness attribute will be translated.
IGNORE	Thickness attribute will be ignored.

Default: IMPLEMENT

- SELECT R-POLYLINE-DISPLAY-POINT {<IGNORE>|USE}

Selects whether or not polyline display points will be used.

IGNORE	Use the control point, ignore display points.
USE	Use display points, ignore the control point.

Default: IGNORE

- SET R-UNITS {<MM> | CM | METER | INCH | FEET }

Sets DXF file units with equivalent **Cimatron** units.

Default: MM

- SELECT R-DIMENSION-AS {<GROUP> | DIMENSION}

Read the dimension as “group” or as “dimension”.

GROUP	Read a dimension from DXF as an INSTANCE.
DIMENSION	Read a dimension from DXF as a dimension.

Default: GROUP

- SELECT R-ATTRIBUTES-AS {<NOTE> | ATTRIBUTES}

Read ATTDEF and ATTRIB entities from DXF either as a Note or as an Attribute

NOTE	Read ATTDEF and ATTRIB entities from DXF as a Note.
ATTRIBUTES	Read ATTDEF and ATTRIB entities from DXF as an Attribute.

Default: NOTE

- SELECT R-DRAFTING-ENTITIES-TO {<MODEL> | VIEW}

  Translate all AutoCAD Model-Space entities to the Cimatron Model or to a dependant view.

  Read the drafting entities as “model” or as “view”.

  MODEL Include all entities, including drafting entities, in the Model. The consequence of this option is that all drafting entities, which could not have been created in the Model (if the file would have been originally created in Cimatron), are in fact translated into the Model.

  VIEW The new pfm file will include a dependant view, named “model”, which will include all drafting entities which will be omitted from the Model.

  Default: MODEL

- SELECT R-PAPER-SPACE {<YES> | NO}

  Translate each AutoCAD layout to a Cimatron drawing.

  YES Each AutoCAD layout will be translated to a Cimatron drawing.

  NO All paper-space entities are ignored during translation.

  Default: YES

Statements for Text Conversion

- MAP TEXT {dxf-font} {cim-font} [{char-map-area-number}]
  MAP R-TEXT {dxf-font} {cim-font} [{char-map-area-number}]

  Maps the source (DXF) style with the equivalent destination (Cimatron) font. Characters from the source font are mapped to destination font characters using the character mapping file (**.cdt** file) and the char-map-area-number. Character mapping files are located in the directory **<root_cad>\dat\fonts**.

  The char-map-area-number is a hexadecimal number indicating the range of source font character codes to be used from the character mapping file when the given source font characters are mapped. Each char-map-area-number specifies a range of 256 characters. The **.cdt** files used by DXF Utilities are:

  rdxf.cdt - for the Read-DXF option

  dxf-font DXF Font Code number of the DXF Font

  cim-font Cimatron Font name, one of the fonts located in the directory **<root_cad>\dat\fonts.**

  char-map-area-number Area number (Hexadecimal number) in the character mapping file (**.cdt** file).

  Defaults:
  If the MAP TEXT statement for a specific font is omitted:
  Default Cimatron font for Read-DXF : standard
  Default DXF style for Read-DXF : standard
  Default char-map-area-number if omitted : 0

- MAP LINE-FONT {DXF-line-font} {cim90-line-font(1-8)}

  Map DXF line-type with equivalent **Cimatron** line-font.

  Defaults (as defined in the default parameter file):

  DXF CONTINUOUS line-type mapped with **Cimatron** line-font 1 (SOLID).

  DXF DASHED line-type mapped with **Cimatron** line-font 4 (L-DASH).

  DXF HIDDEN line-type mapped with **Cimatron** line-font 2 (DASH)

  DXF CENTER line-type mapped with **Cimatron** line-font 1 (SOLID).

  DXF PHANTOM line-type mapped with **Cimatron** line-font 1 (SOLID).

  DXF DOT line-type mapped with **Cimatron** line-font 5 (FONT-5).

  DXF DASHDOT line-type mapped with **Cimatron** line-font 3 (CENTER).

  DXF BORDER line-type mapped with **Cimatron** line-font 1 (SOLID).

  DXF DIVIDE line-type mapped with **Cimatron** default line-font 1 (SOLID).

  Notes:
  - The defaults are set for AutoCAD standard line-types.
  - Other line-types are mapped with **Cimatron** line-font 1 (SOLID).

- MAP COLOR {dxf-color (1-255)} {cim90-color (1-15)}

  Map DXF color with equivalent **Cimatron** color.

  Defaults (as defined in the default parameters file):

  DXF red color mapped with **Cimatron** default color 2.

  DXF yellow color mapped with **Cimatron** default color 3.

  DXF green color mapped with **Cimatron** default color 7.

  DXF cyan color mapped with **Cimatron** default color 5.

  DXF blue color mapped with **Cimatron** default color 4.

  DXF magenta color mapped with **Cimatron** default color 6.

  DXF white color mapped with **Cimatron** default color 1.

  Note:
  - The defaults are set for AutoCAD predefined colors. Other colors are mapped with **Cimatron** color 1.

- MAP PEN {dxf-color (1-255)} {cim90-pen (1-8)}

  Map DXF color with **Cimatron** pen.

  Default: DXF color will be mapped with **Cimatron** pen 1.

Write-DXF Parameters

The parameter file (**.dpr**) contains parameters that control the operation mode, determine the interaction when running Write-DXF, and provide information needed by the utility. These parameters will be described here. The conventions used to describe the parameters are given in the preface.

The user filter parameters are described separately in the section **Selective Translation of Cimatron Files to DXF Files** in this chapter.

Operation Mode Parameters

- SELECT W-MODE {<ASK> | INTER | BATCH}

  The parameter W-MODE selects the mode of operation.

  Write-DXF may be run interactively (INTER) or in batch mode (BATCH).

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. The user is prompted for the name of each **Cimatron part file to be converted.**
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.

  Default: ASK

- SELECT W-LOG-FILE {YES | NO | <ASK>}

  The parameter LOG-FILE selects log file mode.

  Translation process data can either be saved to a log file or displayed on the screen.

YES	All translation data will be saved to a log file. The name of the log file will be that of the DXF file with the extension **.wdl**. The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.
NO	Translation data will be displayed on the screen.
ASK	User is asked whether he wants to save translation data to log file.

  Default: ASK

- SELECT W-SOLID-EXPLODE-AS [SURFACES | BOUNDARIES | <BOTH>}

  Explode SOLID model to surfaces, boundaries, or both.

  Default: BOTH

- SET W-PRECISION { PRECISION (1-9) <6>}

  Number of precision digits.

  Default: 6

Additional Statements

- SET W-TOL {approximation-tolerance (0.001-100) <0.1>}

  The parameter W-TOL sets the tolerance in mm used when approximating a **Cimatron** entity by a DXF entity.

  approximation-tolerance User-defined tolerance; the range is 0.001 - 100 mm.

  Default: 0.1

- SELECT W-DIMENSION-AS {<GROUP> | DIMENSION}

  Write dimensions as a “group” or “dimension”.

  GROUP Write a dimension from **Cimatron** as an INSTANCE into DXF.

  DIMENSION Write a dimension from **Cimatron** as a dimension into DXF.

  Default: GROUP

- SELECT W-FILE-AS {AUTOCAD12 | AUTOCAD13 | <AUTOCAD14> | AUTOCAD2000}

  Save the file in AutoCAD 12, 13, 14 or 2000 format.

  Default: AUTOCAD14

- SELECT W-3D-VIEW-AS {2D | <3D>}.

  View as 2D or 3D and write as 3D.

  When 2D is selected:

  1. For each View

     All translated entities are translated onto one plane - the original view’s rotation, with depth = 0 (plane P).

     Entities of classes 0-2 are projected to the plane P.

     Drafting entities (classes 5-7), which are on planes parallel to plane P are moved to it.

     All other entities - drafting entities on planes which are not parallel to plane P (which means they are meaningless in a 2D view) and surfaces are not translated.

  2. For each Drawing

     Entities belonging to the drawing are translated as explained above.

     Subviews - are written on the original view’s plane (with depth = 0), and are rotated when inserted to the drawing’s plane, with the insertion point moved by the view’s base-point depth, in order to get all subviews on the same plane. (This means that when reading such a drawing back to Cimatron, the product will not be on one plane, since each subview is rotated once again).

     The plane of the original view is identified through examination of the translated current view. Therefore, an attempt to write more than one view per file, with the 2D parameter, will fail.

  Default: 3D

- SELECT W-EXTERNAL-EXPLODE {YES | <NO>}

  Explode external instances.

  Default: NO

- SELECT W-NOTE-AS {TEXT | <MTEXT>}

  Write notes as AutoCAD TEXT or AutoCAD MTEXT.

  AutoCAD has two different text entities; TEXT, which is a one line text entity, and MTEXT, which is a multi-line text entity.

  TEXT Plain text without any styling. The most significant quality of this entity type is that the text can be fixed between two points (and the width of the characters is changed accordingly when displayed).

  MTEXT Creates paragraphs that fit within a non-printing text boundary and it supports style definiitons.

  There is no way to keep both text length and style when translating to AutoCAD.

  Default: MTEXT

- SELECT W-LEVEL-MASK {<ALL ON> | AS IS}

  Set the Levels Visibility as in the Cimatron part file or set all levels on.

  ALL ON View all levels of the part file.

  AS IS View the levels as defined in the Cimatron part file.

  Default: ALL ON

- SELECT W-APPROXIMATE-CURVE {<YES> | NO}

  Approximate Cimatron curves.

  YES Approximate all Cimatron curves to polygons (default).

  NO Translate Cimatron ellipse entities to AutoCAD ellipses and Cimatron spline entities (2D or 3D) to AutoCAD splines.

  Since both entity types were added to AutoCAD in version 13, if the DXF or DWG file is created in AutoCAD version 12 (by choosing the parameter **Save File As AutoCAD12**), the W-CRV-TO-POLYGON parameter is ignored (and all curves are approximated to polygons).

  Default: YES

- SELECT W-SUPPRESSED-PARTS {<IGNORE> | TRANSLATE}

  Select the appropriate option for translating suppressed parts.

  IGNORE Ignore the “suppressed” attribute of the assembly parts (and translate them as usual to DXF).

  TRANSLATE Do not translate the suppressed parts.

  Default: IGNORE

- SELECT W-SUBVIEW-INSTANCE-EXPLODE {YES | <NO>}

  Choose how to translate Subview Instances in drawings.

  YES Explode them and translate them as entities belonging to the drawing.

  NO Group them and translate them as inserts.

  Default: NO

Statements for Text Conversion

- MAP TEXT {dxf-font} {cim-font} [{char-map-area-number}]
  MAP W-TEXT {dxf-font} {cim-font} [{char-map-area-number}]

  Maps the source (**Cimatron**) font with the equivalent destination (DXF) font. Characters from the source font are mapped to destination font characters using the character mapping file (**.cdt** file). Character mapping files are located in the directory **<root_cad>\dat\fonts**. The **.cdt** files used by DXF Utilities are:

  tright.cdt and **tleft.cdt** - for the Write-DXF option

  dxf-font DXF Font name

  cim-font **Cimatron** Font name, one of the fonts located in the directory **<root_cad>\dat\fonts**.

  Defaults:
  If the MAP TEXT statement for a specific font is omitted:
  Default **Cimatron** font for Read-DXF : standard
  Default DXF font for Write-DXF : standard

- MAP COLOR {dxf-color (1-255)} {cim90-color (1-15)}

  Map a **Cimatron** color with the equivalent DXF color.

  Defaults (as defined in the default parameters file):

  Cimatron default color 2 mapped with DXF red color (1).

  Cimatron default color 3 mapped with DXF yellow color (2).

  Cimatron default color 7 mapped with DXF green color (3).

  Cimatron default color 5 mapped with DXF cyan color (4).

  Cimatron default color 4 mapped with DXF blue color (5).

  Cimatron default color 6 mapped with DXF magenta color (6).

  Cimatron default color 1 mapped with DXF white color (7).

  Note:
  - The defaults are set for AutoCAD predefined colors. Other colors are mapped with AutoCAD white color (7).

- MAP PEN {dxf-color (1-255)} {cim90-pen (1-8)}

  Map a **Cimatron** pen with the equivalent DXF pen. In the event of a conflict between color mapping and pen mapping, pen mapping overrides the color mapping.

  Default: None.

Selective Translation of Cimatron Files to DXF Files

Selective writing allows selective translation of data from the **Cimatron** database to a file in the DXF format. This is accomplished by applying user-defined filters which control whether or not a particular entity is written to the DXF file.

Selective translation according to application filters is especially important, since a DXF file can only include one **Cimatron** drawing or view. By using an application filter, the specific drawing, view or only the model can be selected for translation.

User Input

Figure 5-1 shows the influence of user-defined filters.

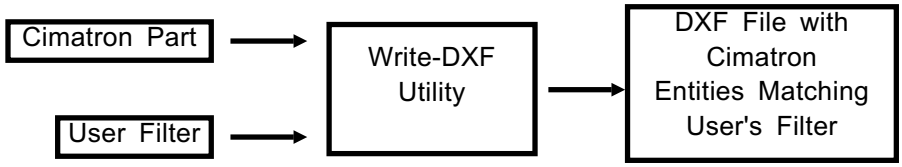


Figure 5-1: Influence of User Defined Filters on the Write-DXF Utility

By default, the **Write-DXF** utility has no filters. All entities in the given part file are translated to the DXF file.

User-defined filter statements are placed in the parameter file (**.dpr**). Two types of filters are available:

- Application:
  model, view, drawing
- Attribute:
  level, line-font, color, pen, blank, set

A filter causes explicit and implicit inclusion or exclusion of corresponding values. Explicit inclusion/exclusion is the direct result of the filter. Implicit inclusion/exclusion results from the influence of *specified* argument values on argument values that were *not specified* in the definition.

During processing, explicit inclusion and exclusion always override implicit inclusion and exclusion. Application group filters and attribute group filters are processed separately.

Filter Syntax

Filter statements have the following syntax:

SELECT FILTER-TYPE FILTER-ARGUMENT ARGUMENT-VALUE

For example, in the filter statement SELECT W-APPLICATION DRAWING D1, the filter type is APPLICATION, the filter argument is DRAWING and the argument value is D1.

Application Filters

Application filters allow entities from specific application(s) to be included.

Following is the syntax of the application filter definition statement:

```
SELECT W-APPLICATION [ SPLIT | NOT] {MODEL}
SELECT W-APPLICATION [ SPLIT | NOT] {VIEW [view-name(s)]}
SELECT W-APPLICATION [ SPLIT | NOT] {DRAWING [drawing-name(s)]}
```

Note that the filter argument MODEL takes no argument values.

Entities are included according to their visibility in the application. For example, an entity that is visible in an included view will be translated into the DXF file, even if the model in which the entity was created is not included. To give another example, including a drawing causes all views referenced within it to be written to the file.

The following rules apply:

- When no argument values are given, all available argument values for the specified application are included, and all other filter arguments are implicitly excluded.

  For example: SELECT W-APPLICATION VIEW means that all defined views in the part-file database are included, while the model and all drawings are excluded.

- When argument values are given, the specified argument values of the application are included. Other argument values for this filter argument, and other filter arguments, are implicitly excluded.

  For example: SELECT W-APPLICATION VIEW view1 view2 means that view1 and view2 are included. Other views, the model, and all drawings are excluded.

- Complement filter arguments, which begin with NOT, have different meanings according to the context.

  - When no argument values are given, all possible argument values for the specified filter argument are excluded. All other filter arguments are implicitly included.

    For example: SELECT W-APPLICATION NOT DRAWING means that all drawings are excluded, and model and all views are included.

  - When argument values are specified, all other argument values for the specified filter argument are included. The specified argument values, and all other applications are implicitly excluded.

    For example: SELECT W-APPLICATION NOT VIEW view1 view2 means that all views other than view1 and view2 are included, and view1, view2, the model and all drawings are excluded.

- The use of DRAWING and VIEW filters simultaneously may create a conflict. For example, when a drawing is included and one of the views within the drawing is excluded, a conflict occurs. To prevent such conflicts, the DRAWING filter and its implicit influence override the VIEW filter's explicit and implicit influence.

The effect of various filter argument and argument value combinations with the APPLICATION filter is summarized in Table 5-2.

Table 5-2. APPLICATION Filter: Explicit and Implicit Inclusion and Exclusion of Entities for Various Filter Arguments and Argument Values

Filter Argument and Argument Values	Included	Excluded
MODEL	model	all views all drawings
VIEW	all views	model * all drawings
VIEW V1 ... Vn	views V1 ... Vn	model * views not in V1 ... Vn all drawings
DRAWING	all drawings	model ** all views **
DRAWING D1 ... Dn	drawings D1 ... Dn	model all views drawings not in D1 ... Dn
NOT MODEL	all views * all drawings	model
NOT VIEW	model all drawings	all views
NOT VIEW V1 ... Vn	views not in V1 ... Vn	model * views V1 ... Vn all drawings
NOT DRAWING	model ** all views **	all drawings
NOT DRAWING D1 ... Dn	drawings not in D1 ... Dn	model ** all views ** drawings D1 ... Dn

Note:

- Entities are included according to their existance in the application. For example, an entity that is visible in an included view will be translated into the DXF file, even if the model in which the entity was created is not included.

- Split the PFM file to many output files. This option enables you to write the model, each view, or each drawing of a PFM file to a different file.
  The output files are placed in a directory named after the PFM file.
  The view/drawing file name prefix is identical to the translated view/drawing name with an addition (_d for drawing, _v for views).
  The model file name prefix is model. If the name of the view or drawing contains special characters, such as a dot (.), dollar sign ($), etc., they will be converted to an underscore (_) in the output file name.

* All entities that were deleted or changed in the views.

** All entities that were deleted or changed in the drawing.

Attribute Filters

Attribute filters allow entities with specified attributes to be included. For example, specifying a color will cause entities with this color to be included.

Following is the syntax of the attribute filter definition statement:

```
SELECT W-ATTRIBUTES [NOT] {LEVEL level-name(s)}
SELECT W-ATTRIBUTES [NOT] {LINE-FONT line-font-name(s)}
SELECT W-ATTRIBUTES [NOT] {COLOR color-number(s)}
SELECT W-ATTRIBUTES [NOT] {PEN pen-numbers(s)}
SELECT W-ATTRIBUTES [NOT] {BLANK}
SELECT W-ATTRIBUTES [NOT] {SET SURFACES}
```

The following rules apply:

- Argument values must always be given. The specified argument values are included, and all other argument values for the filter argument are excluded. If an attribute filter statement is given without argument values, such as SELECT W-ATTRIBUTES LEVEL, an error message will appear.
- Complement filter arguments, which begin with NOT, cause exclusion of the specified argument values and implicit inclusion of all other possible argument values.

  For example: SELECT W-ATTRIBUTES NOT LINE-FONT SOLID DOTTED means inclusion of all line-fonts other than SOLID or DOTTED.
- Line-font-name refers to the names of line-fonts displayed on the **Cimatron** line-font menu: SOLID, DASH, CENTER, L_DASH, FONT_5, FONT_6, FONT_7, FONT_8 (refer to the LINATT function).
- Color-number refers to the position of the color in the **Cimatron** color menu (refer to the FILE >> SETUP >> COLOR SETUP function). The range of color numbers is 1 to 15, where the first (top) entry in the menu has a color-number of 1. For example, if red is the third entry in the color menu, then the color-number of red is 3.
- Pen-number refers to the pen numbers displayed in the **Cimatron** pen menu (refer to the LINATT function). The range of pen numbers is 1 to 8.
- The BLANK filter is useful when blanked entities are not needed in the output DXF file. SELECT W-ATTRIBUTES NOT BLANK will cause the **Write-DXF** utility to ignore blanked entities.
- The SET filter is actually a filter for classes of entities in **Cimatron**. *SET* is a predefined collection of classes. Currently, the only available set is the SURFACES set which consists of all **Cimatron** surface types.

Logical Operation Rules

Figure 5-2 shows the use of logical operations with filter statements.

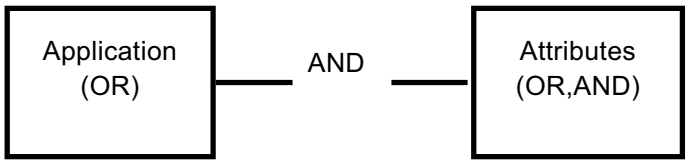


Figure 5-2: The Use of logical Operations with Filter Statements

Figure 5-3 gives examples of the logical operation rules as applied to various filter statements. These examples will be referred to in the following explanations.

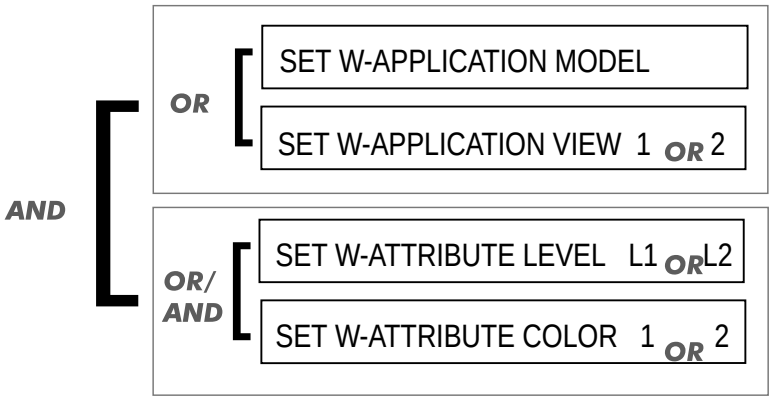


Figure 5-3. Examples of Logical Operations Applied to Filter Statements

The OR operation is performed on all argument values of the same filter argument. For example, in the statement MSET W-APPLICATION VIEW 1 2, the OR operation is applied. This results in translation of entities that are visible in view 1, or visible in view 2, or visible in both views. If there is more than one statement with the same filter argument, the argument values from the different statements are treated as if they are all in one statement.

The OR operation is also applied between application filters which have different filter arguments, as shown in Figure 5-3. For example, defining the MODEL and VIEW filter statements shown in Figure 5-3 will cause translation of entities visible in the model, or visible in views 1 or 2, or visible both in the model and in views 1 and 2.

By default, the AND operation is applied between different types of attribute filters. In order for an entity to be translated, it must match at least one argument value, in each of the attribute filter statements.

To set the default to the OR operation, specify SELECT W-ATTRIBUTES FILTER OR in the parameters file. The format of the statement is:

SELECT W-ATTRIBUTES {FILTER {<AND> | OR }}

The AND operation is applied between filters of different types, i.e. between application and attribute filters.

Number of Precision Digits

This specifies the number of precision digits after the decimal point.

```
SET W-PRECISION {precision (1-9) <6>}
```

Automatically explode SOLID objects/entities when writing a SOLID Part File to DXF. ASSEMBLY part-file handling is not in the scope of this project.
The function will have the possibility to automatically divide the output to two different DXF files: one for the SOLID model and the other for the SOLID drafting application, i.e. VIEWS, SECTION, and DRAWINGS.

The directory, named according to the user input, contains DXF files. The model is written to a file named **model.DXF**. The drafting applications are written to a file named **draft.DXF**. If the SPLIT option was selected by the user, then drafting applications are written to different files named following the current SPLIT rules. A log-file is created for each DXF file. The log file will include a remark that the file came from the SOLID pfm.

```
SELECT W-SOLID-EXPLODE-AS {SURFACES | BOUNDARIES | <BOTH>}
```

Write dimension as “group” or as “dimension”.

```
SELECT W-DIMENSION-AS {<GROUP>  | DIMENSION}
```

Set levels mask as in Cimatron part file, or set all levels ON.

```
SELECT W-LEVEL-MASK {<ALL-ON>  | AS-IS}
```

Statements for Conversion

Map DXF font with equivalent **Cimatron** font and map the character area.

```
MAP TEXT {DXF-font} {cim-font} [{char-map-area-number}]
MAP R-TEXT {DXF-font} {cim-font} [{char-map-area-number}]
MAP W-TEXT {DXF-font} {cim-font} [{char-map-area-number}]
```

Map DXF color with equivalent **Cimatron** color.

```
MAP COLOR {DXF-color (1-255)} {cim90-color (1-15)}
```

Map DXF color with equivalent **Cimatron** pen.

```
MAP PEN {DXF-color (1-255)} {cim90-pen (1-15)}
```

Map DWG line-type with equivalent **Cimatron** line-font.

Selective Translation Statements for Write-DXF Utility

These optional statements allow selective translation of **Cimatron** entities to DXF entities. For further information, refer to the section **Selective Translation of Cimatron Files to DXF Files** in Chapter 4.

```
SELECT W-APPLICATION [SPLIT | NOT] {MODEL}
SELECT W-APPLICATION [SPLIT | NOT] {VIEW [view-name(s)]}
SELECT W-APPLICATION [SPLIT | NOT] {DRAWING [drawing-name(s)]}
SELECT W-APPLICATION [NOT] {LEVEL level-name(s)}
SELECT W-APPLICATION [NOT] {LINE-FONT line-font-name(s)}
SELECT W-APPLICATION [NOT] {COLOR color-number(s)}
SELECT W-APPLICATION [NOT] {PEN pen-number(s)}
SELECT W-APPLICATION [NOT] {BLANK}
SELECT W-APPLICATION [NOT] {SET SURFACES }
SELECT W-APPLICATION {FILTER {<AND> | OR}}
```

USE FILE Statement - Nested Parameter File

This statement instructs the program to include a file which contains parameter statements during the processing of the parameter (**.dpr**) file. This enables the user to nest files of parameter statements.

The statement syntax is:

```
USE FILE {file-pathname}
```

- File-pathname is the full absolute file name in the file system of the included file. There are no limitations on the name, extension or location of the included file.
- The included file is processed as soon as the USER FILE statement is encountered. The rule for multiple occurrences of the same statement is applied as if all processed statements from all nested files are in one parameter file.

As an example of how this feature can be used, consider a multiple user environment. Since certain information should be the same for each user, a system parameter file that defines all the global system parameters can be written. At the same time, certain information will be specific to each user. To meet both needs, the user can create his own parameter file which will contain user-specific information as well as a USE FILE statement to include the system parameter statements.

Default Parameter File and Parameter Values

Following is the default parameter file:

```
SELECT R-MODE ASK
SELECT R-LOG-FILE ASK
SELECT R-DIMENSION-AS GROUP
SELECT R-ATTRIBUTES-AS NOTE
SELECT R-THICKNESS IMPLEMENT
SELECT R-POLYLINE-DISPLAY-POINTS IGNORE
SELECT R-UNITS MM
SELECT R-DRAFTING-ENTITIES-TO MODEL
SELECT R-PAPER-SPACE
SELECT W-MODE ASK
SELECT W-LOG-FILE ASK
SELECT W-SOLID-EXPLODE-AS BOTH
SELECT W-DIMENSION-AS GROUP
SELECT W-FILE-AS AUTOCAD14
SELECT NOTE_AS NTEXT
SELECT W-3D-VIEW-AS 3D
SELECT W-EXTERNAL-EXPLODE NO
SELECT CRV-TO-POLYGON YES
SELECT W-LEVEL-MASK ALL-ON
SELECT W-IGNORE-SUPPRESSED-PARTS NO
SET W-TOL 0.1
SET W-PRECISION 6
MAP LINE-FONT CONTINUOUS    1
MAP LINE-FONT DASHED        4
MAP LINE-FONT HIDDEN        2
MAP LINE-FONT CENTER        1
```

```
MAP LINE-FONT PHANTOM      1
MAP LINE-FONT DOT          5
MAP LINE-FONT DASHDOT      3
MAP LINE-FONT BORDER       1
MAP LINE-FONT DIVIDE       1
MAP COLOR  1               2
MAP COLOR  2               3
MAP COLOR  3               7
MAP COLOR  4               5
MAP COLOR  5               4
MAP COLOR  6               6
MAP COLOR  7               1
```

Parameter File Check

The DXF file utilities provides an option to check the format and statement syntax of the parameter file.

To check a file, use option **PARAMETER FILE CHECK**, from the DXF Utilities menu. If no errors are found in the parameter file, the following message will appear:

```
PARAMETER FILE WAS SUCCESSFULLY PARSED
```

The parameter file check does not verify statement arguments. ❑

Chapter 6
DXF File Information Utility

The DXF File Information Utility provides the user with a brief description of the DXF file and a report on its validity. The utility reads the whole DXF file and generates a report file (**.drt**). The report file includes three sections:

1. Reading file section includes any syntax or structural errors found in the file.
2. Layers, Linetype and Style tables section lists these tables from the DXF file together with some of their important attributes. This data can be helpful for the user when mapping DXF Linetype and Style with their **Cimatron** equivalents via the DXF parameters file.
3. Entities statistics sections display the types and quantity of entities in the DXF file.

Running the DXF File Information Utility

The DXF File Information Utility may be run from the graphic or textual interaction.

Graphic

How To:

1. Select Application.
2. Select DXF.
3. Select DXF Info.

Textual

How To:

1. Select option DXF FILE INFORMATION.
2. Type the name of the DXF file when prompted.
3. Messages will be displayed as the report file (.drt) is completed.

Notes:

- The report file created will be in the same directory and have the same name as the DXF file, but will have the extension **.drt** rather than **.dxf**.
- The DXF file may be located in the current directory or in other directories. Enter the full pathname of the DXF file if it is not located in the current directory.

Sample DXF Report File

```
  Cimatron  (version  11.0)  DXF  File  Information  <Version  2000
==================================================================
Reading  DXF  File  :
_________
End  of  reading  DXF  file  DXF  File  Tables  :
__________-

#  Layers  table     :
        Layer  Name          Color          Linetype              Visibility
        ______          ___-          _____          ______                    0
7            CONTINUOUS              ON                                  1
8            CONTINUOUS              ON
AME_FRZ                 7            CONTINUOUS              ON
          DEFPOINTS              7            CONTINUOUS              ON

#  Ltype  table       :

        Linetype                    Ltype  Description
        ____                     _________
        CONTINUOUS                   Solid  line
        PHANTOM                       ______  __  __  ___

#  Style  table       :

        Style  Name                 Font  File  Name         Special  Form
        ______-                  _______-          _______
          HEBREW                        HEBREW                    NO
          ROMANS                        romans                    NO
          STANDARD                      txt                       NO

DXF  File  Contains  the  Following  Data:
_____________________-
ARC                 NUMBER  OF  ENTITIES  :  85
CIRCLE              NUMBER  OF  ENTITIES  :  7
LINE                NUMBER  OF  ENTITIES  :  323
POINT               NUMBER  OF  ENTITIES  :  170
BLOCK               NUMBER  OF  ENTITIES  :  23
SOLID               NUMBER  OF  ENTITIES  :  111
TEXT                NUMBER  OF  ENTITIES  :  155
INSERT              NUMBER  OF  ENTITIES  :  11
DIMENSION           NUMBER  OF  ENTITIES  :  11      ❑
```

Chapter 7
PFM File Information Utility

The PFM File Information Utility provides you with a brief description of the PFM file. The utility reads the PFM file and generates a report file (**.pfr**).

The report includes:

1. Names of levels within the PFM file.
   ON = The level is visible in the modal.
   OFF = The level is not visible in the modal.
2. Names of views within the PFM file.
3. Names of drawings within the PFM file.
   Names of views included in each drawing.
4. Entity statistics sections display the types and quantities of entities in the PFM file.

Running the PFM File Information Utility

The PFM File Information Utility may be run from the graphic or textual interaction.

Graphic

How To:

1. Select Application.
2. Select DXF.
3. Select PFM Info.

Textual

How To:

1. Select option PFM FILE INFORMATION.
2. Type the name of the PFM file when prompted.
3. Messages will be displayed as the report file (**.pfr**) is completed.

Notes:

- The report file created will be in the same directory and have the same name as the PFM file, but will have the extension **.pfr** rather than **.pfm**.
- The PFM file may be located in the current directory or in other directories. Enter the full pathname of the PFM file if it is not located in the current directory.

```
Cimatron  (version  11.0)  PFM  File  Information  (Version  2000)
================================================================
PFM  File  Contains  the  Following  Levels:
____________________-
1          -  ON            2         -  OFF            3          -  OFF
4          -  OFF           5         -  OFF            6          -  OFF
7          -  OFF           8         -  OFF            9          -  OFF
10         -  OFF           11        -  OFF            12         -  OFF
13         -  OFF           14        -  OFF            15         -  OFF
16         -  OFF           17        -  OFF            18         -  OFF
19         -  OFF           20        -  OFF            21         -  OFF
22         -  OFF           23        -  OFF            24         -  OFF
25         -  OFF           26        -  OFF            27         -  OFF
28         -  OFF           29        -  OFF            30         -  OFF
31         -  OFF           32        -  OFF            33         -  OFF
34         -  OFF           35        -  OFF            36         -  OFF
37         -  OFF           38        -  OFF            39         -  OFF
40         -  OFF           41        -  OFF            42         -  OFF
43         -  OFF           44        -  OFF            45         -  OFF
46         -  OFF           47        -  OFF            48         -  OFF
49         -  OFF           50        -  OFF            WRK        -  OFF


PFM  File  Contains  the  Following  Views:
____________________-
            V1
            V2
            V3
            V4

PFM  File  Contains  the  Following  Drawings:
____________________
            D1

Views  included  in  drawing:
            V2
            V1
            V3
            V4

PFM  File  Contains  the  Following  Data:
____________________-
LINE                      Class  1     Type  1     Number  of  Entities  :  33
ARC                       Class  1     Type  2     Number  of  Entities  :  6
LINEAR DIMENSION          Class  5     Type  1     Number  of  Entities  :  5
RADIAL DIMENSION          Class  5     Type  2     Number  of  Entities  :  3
ANGULAR DIMENSION         Class  5     Type  3     Number  of  Entities  :  3
ORDINAT DIMENSION         Class  5     Type  4     Number  of  Entities  :  9
SECTION LINE              Class  6     Type  1     Number  of  Entities  :  2
HATCH                     Class  6     Type  2     Number  of  Entities  :  1
ID_NUM                    Class  6     Type  7     Number  of  Entities  :  3
INSTANCE                  Class  13    Type  1    Number  of  Entities  :  4
```

❑

Section III

IGES Utilities

Chapter 8
Introduction to the IGES Utilities

Part files, including geometric entities, dimensions and kinds of annotation, levels, views, masters and their instances, may all be transferred between **Cimatron** and other systems in the IGES format, approved by the National Institute of Standards and Technology (N.I.S.T.) of the U.S. Department of Commerce.

The following five functions are included in the **Cimatron** IGES Utilities:

Read-IGES	Convert files that are in IGES format to files in **Cimatron** part file **(.pfm)** format.
Write-IGES	Convert **Cimatron** part files **(.pfm)** to files in IGES format.
PARAMETER FILE CHECK	Verify an IGES parameter file **(.ipr)**.
IGES FILE INFORMATION	Generate a report **(.drt)** on the IGES file tables and entities, and any syntax or structural errors in the file.
PFM FILE INFORMATION	Generate a report **(.pfr)** file on the IGES file tables and entities, and any syntax or structural errors in the file.

Running IGES

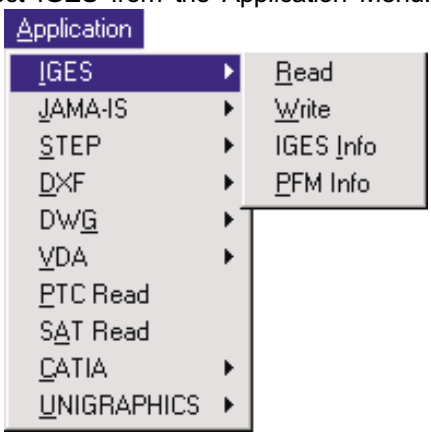
There are two methods of running the IGES Data Interface Utilities:

- **graphically**
- **textually**

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select IGES from the Application Menu. The following options are displayed:



3. Select the appropriate option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following IGES options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
4  ->  IGES  File  Information  option
5  ->  PFM  File  Information  option
```

3. Select the appropriate option.

Each of these utilities will be explained in a chapter of this section.

About the Initial Graphics Exchange Specification (IGES)

What is IGES?

CAD/CAM systems are in use today in increasing numbers for applications covering all phases of the design, analysis, manufacturing and testing of products. As users in each of these stages may use systems supplied by different vendors, there is a need for data to be exchanged digitally between these systems.

IGES, the Initial Graphics Exchange Specification, answers this need by defining a neutral data format that enables the digital exchange of information between CAD/CAM systems.

How Does it Work?

The databases of CAD/CAM systems from different vendors are often incompatible with one another. For example, an arc on one system may be defined by its center, its diameter start angle and end angle, while on another system an arc may be defined by a center point, start point and end point. IGES solves this problem by providing a common format for the digital exchange of database information that is independent of all CAD/CAM systems.

Translators convert a proprietary internal database format into a neutral IGES format and from the IGES format into another vendor's internal database. These translators (pre- and post-processors), are normally supplied as part of a vendor's product.

Overview of the IGES File Format

This section contains a quick overview of the IGES file format currently supported by Cimatron. For more information, refer to the IGES standard itself.

Cimatron supports only the fixed-length ASCII (American Standard Code for Information Interchange) format, which stores information in records of 80 characters. This format can be read and modified by any ASCII text editor.

Figure 8-1 shows a modeling, view and drawing file (line.pfm) used to generate the sample ASCII IGES file (line.igs) shown in Figure 8-2, via the **Cimatron** command Write-IGES.

Note:

- In Figure 8-1, the line begins at the origin of the UCS (User Coordinate System) and has a length of 100. The color of the line is red and the color of the dimension line is white.

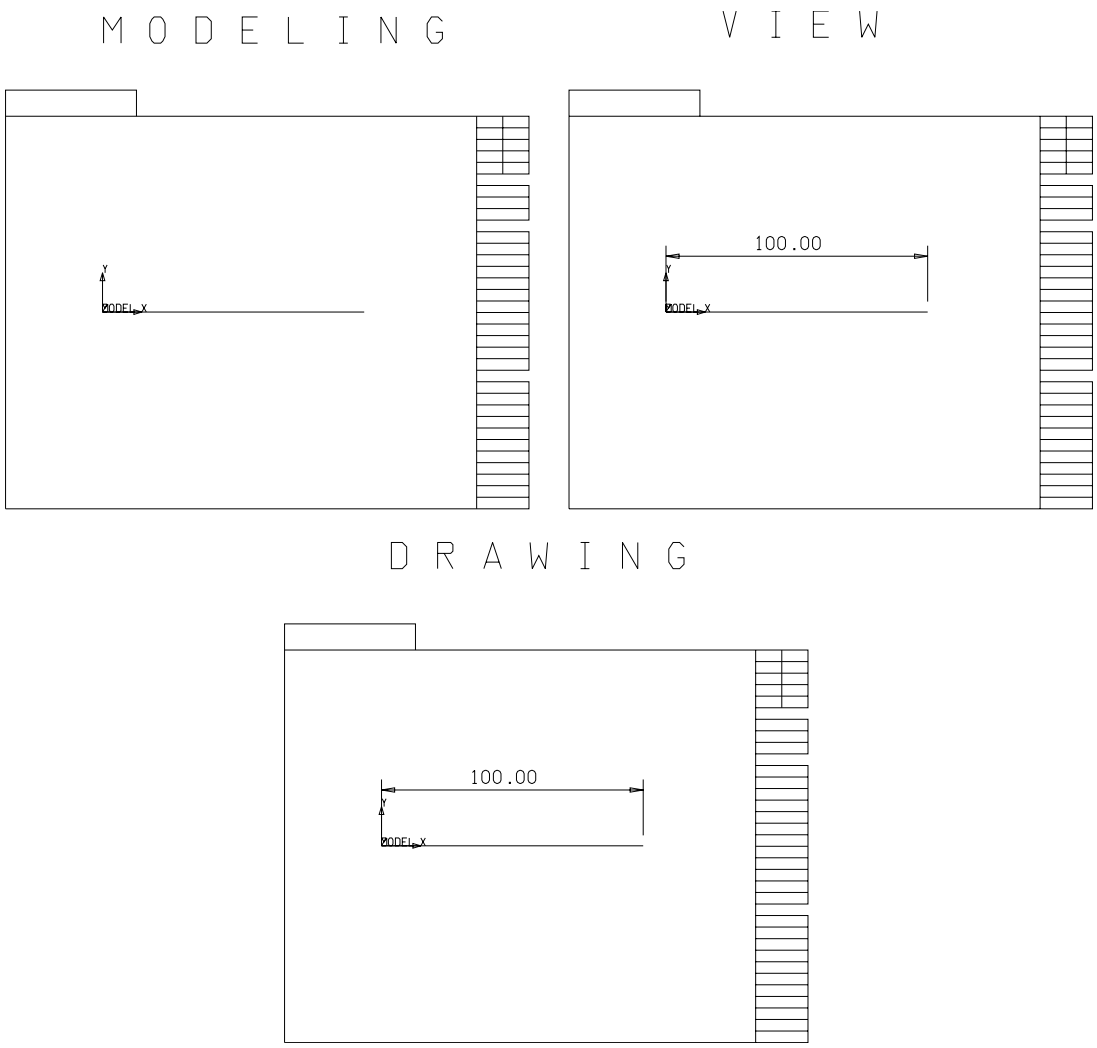


Figure 8-1: Model, View and Drawing (line.pfm) Used to Generate the IGES File

IGES File Structure

```
Cimatron Iges convertor                                                    S   1
H,,1H;,4HNONE,27HE:\UDD\user\test\angle2.igs,14HCimatron V9.0,40HCimat     G   1
ron IGES (V5.2) Convertor V5.0.6.47,16,24,8,56,8,4HNONE,1.00000,2,2HMM,8   G   2
,0.400000,13H961126.100021,0.000100000,10000.0,4HUser,8HCimatron,10,0,13H961 G   3
126.092724;                                                                G   4
     314      1      0      0      0      0      0                  000000000D   1
     314      0      0      1      0                                        0D   2
     314      2      0      0      0      0      0                  000000000D   3
     314      0      0      1      0                                        0D   4
     314      3      0      0      0      0      0                  000000000D   5
     314      0      0      1      0                                        0D   6
     314      4      0      0      0      0      0                  000000000D   7
     314      0      0      1      0                                        0D   8
     314      5      0      0      0      0      0                  000000000D   9
     314      0      0      1      0                                        0D  10
     314      6      0      0      0      0      0                  000000000D  11
     314      0      0      1      0                                        0D  12
     314      7      0      0      0      0      0                  000000000D  13
     314      0      0      1      0                                        0D  14
     314      8      0      0      0      0      0                  000000000D  15
     314      0      0      1      0                                        0D  16
     212      9      0      1      0      0      0                  000010100D  17
     212      1      8      2      0                                        0D  18
     110     11      0      1      0      0      0                  000010100D  19
     110      1      5      2      0                                        0D  20
     212     13      0      1      0      0      0                  000010100D  21
     212      1      2      2      0                                        0D  22
     212     15      0      1      0      0      0                  000010100D  23
     212      1      7      2      0                                        0D  24
     212     17      0      1      0      0      0                  000010100D  25
     212      1      6      2      0                                        0D  26
     212     19      0      1      0      0      0                  000010100D  27
     212      1      5      2      0                                        0D  28
     110     21      0      1      0      0      0                  000010100D  29
     110      1      6      2      0                                        0D  30
     110     23      0      1      0      0      0                  000010100D  31
     110      1      8      2      0                                        0D  32
     110     25      0      1      0      0      0                  000010100D  33
     110      1      2      2      0                                        0D  34
     110     27      0      1      0      0      0                  000010100D  35
     110      1      7      2      0                                        0D  36
     404     29      0      0      0      0      0                  000000100D  37
     404      0      0      1      0                                        0D  38
314,94.11765,100.,74.5098,4Hcol1,0;                                        1P   1
314,11.7647,60.78431,0.,4Hcol2,0;                                          3P   2
314,100.,94.11765,70.58824,4Hcol3,0;                                       5P   3
314,82.35295,7.843137,0.,4Hcol4,0;                                         7P   4
314,100.,100.,90.19608,4Hcol5,0;                                           9P   5
314,23.52941,23.52941,78.43137,4Hcol6,0;                                  11P   6
314,94.11765,94.11765,66.66666,4Hcol7,0;                                  13P   7
314,100.,0.,39.21569,4Hcol8,0;                                            15P   8
212,1,8,61.2541,10.23737,1,1.570796,0.,0,0,-6.923902,-17.59825,            17P   9
0.,8Hangale=0,0;                                                          17P  10
110,-60.260904673567,39.152565183391,0.,-60.260904673567,                 19P  11
101.15256518339,0.,0;                                                     19P  12
212,1,8,61.2541,10.24738,1,1.570796,.5235988,0,0,-28.26035,               21P  13
25.02218,0.,8Hangle=30,0;                                                 21P  14
212,1,9,68.05399,10.25738,1,1.570796,3.141593,0,0,-109.297,               23P  15
.
.
S        1G        4D       38P       29                                   T    1
```

Figure 8-2: Sample ASCII IGES File

The IGES file is a fixed-length ASCII file with five required sections. The sections must always appear in the same order, as presented here.

Start Section

The start section is an unformatted prologue to the IGES file that is used to indicate certain conventions within the file (for example, what type of information is included in a specific group of layers). All records in the section have the letter S in Column 73 and a sequence number in Columns 74 through 80. The information in Column 1 through 72 need not be formatted in any special way except that the ASCII character set must be used.

The IGES specification does not contain any recommendations regarding the use of this section. Note that several CAD systems do not allow the user to write information to this section.

An example of the start section is given below:

```
Cimatron Iges convertor                                    S  1
```

Global Section

The global section contains information such as product identification, number representation, author of the model that the IGES file represents, CAD system and IGES translator version.

All records in the section have the letter G in Column 73 and a sequence number in Columns 74 through 80.

String constants are written in Hollerith form which consists of #Hstring, where # is an unsigned integer which gives the number of characters in string. Blanks, parameter delimiters and record delimiters are counted as characters. For example, in the entry **27HE:\UDD\USER\test\angle2.igs**, there are 27 characters following the H. There is no limit to the number of characters in any field, or the number of lines, however ASCII control characters cannot be used.

The global section of the IGES file given in Figure 8-2 is shown below. An explanation of the IGES fields follows.

```
Cimatron  Iges  convertor                                                S   1
1H,,1H;,4HNONE,27HE:\UDD\USER\test\angle2.igs,14HCimatron V11.0,40HCimat      G  1
ron  IGES  (V5.2)  Convertor  V5.0.6.47,16,24,8,56,8,4HNONE,1.00000,2,2HMM,8  G  2
,0.400000,13H961126.100021,0.000100000,10000.0,4HUser,8HCimatron,10,0,13H961 G  3
126.092724;                                                                  G  4
```

Parameter Delimiter
: `,` (comma)

Separates each field.

Record Delimiter
: `;` (semicolon)

Separates each record.

Product Identification from Sender
: `test`

Reference name or identifier used by the sender.

File Name
: `line.igs`

System ID
: Cimatron V11.0

Preprocessor Version
: Cimatron `Iges Convertor` V5.3

Version of the preprocessor (the preprocessor translates from **Cimatron** format to the neutral IGES format).

Number of Bits for integer
: `16`

The number of bits present in the integer representation of the sending system.

Single Precision Magnitude
: `24`

The maximum power of ten which may be represented as a single precision floating point number on the sending system.

Single Precision Signific.
: `8`

The number of decimal digits of significance which can be accurately represented in the single precision floating point representation on the sending system.

Double Precision Magnitude
: `56`

The maximum power of ten which may be represented as a double precision floating point number on the sending system.

Double Precision Signific.
: `8`

The number of decimal digits of significance which can be accurately represented in the double precision floating point representation on the sending system.

Product Identification for Receiver

TEST

The name used by the receiver to identify the product. This need not be the same as the product identification used by the sender.

Model Space Scale

1.00000

The ratio of model space to real world space.

Unit Flag

2

Flag which sets the type of model units used by the system.

Flag Value	Measuring System
1	Inches
2	Millimeters
3	-
4	Feet
5	Miles
6	Meters
7	Kilometers
8	Mils (i.e.,0.001 inch)
9	Microns
10	Centimeters
11	Microinches

Units

MM

A string constant which sets the model units used by the system.

Constant	Measuring System
IN or INCH	Inches
MM	Millimeters
FT	Feet
MI	Miles
M	Meters
KM	Kilometers
MIL	Mils (i.e.,0.001 inch)
UM	Microns
CM	Centimeters
UIN	Microinches

Line Weight Gradations

8

The number of equal divisions of line thickness (The maximum line width parameter).

Size of Maximum Line Width in Units

`0.400000`

Actual width of the thickest line possible in the (scaled) file.

Date and Time of File Generation

`961126.092724`

yymmdd.hhnnss (y=year, m=month, d=day, h=hour, n=minute, s=second)

Minimum User Intended Resolution

`0.000100000`

The smallest discernible distance in model space units. Coordinate locations in the file which are less than this distance apart should be considered as coincident.

Approximate Maximum Coordinate Value:

`10000.0`

The absolute magnitude of each of the coordinates in this model is less than or equal to this value.

Author `user`

Organization `Cimatron`

Version Number Flag `10`

IGES version number.

Flag	IGES Version
0	No standard specified.
1	1.0
3	2.0
4	3.0
6	4.0
8	5.0
9	5.1
10	5.2

Drafting Standard Code `0 (No standard specified)`

The drafting standard used to generate the data in this file.

Directory Entry Section

The directory entry section has one directory entry for each entity in the file. The entry for each entity is fixed in size and contains twenty fields of eight characters each, spread across two consecutive eighty character lines. The purposes of the Directory Entry Section are to provide an index for the file and to contain attribute information for each entity.

Table 8-1 gives an abbreviated listing of the fields making up the directory entry for each entity.

Table 8-1. Format of the Directory Entry Section in ASCII Form

1 8	9 16	17 24	25 32	33 40	41 48	49 56	57 64	65 72	73 80
(1) Entity Type Number #	(2) Parameter Data →	(3) Structure #,→	(4) Line Font Pattern #,→	(5) Level #,→	(6) View 0,→	(7) Transformation Matrix 0,→	(8) Label Display Assoc. 0,→	(9) Status Number #	(10) Sequence Number D#
(11) Entity Type Number #	(12) Line Weight Number #	(13) Color Number #,→	(14) Parameter Line Count #	(15) Form Number #	(16) Reserved	(17) Reserved	(18) Entity Label	(19) Entity Subscript Number #	(20) Sequence Number D# + 1

Nomenclature:

```
(n)  Field  Number  n
#    Integer
→   Pointer
#,→ Integer  or  pointer  (pointer  has  negative  sign)"Symbol"M
0,→ Zero  or  pointer
```

For example:

```
 212      9      0      1      0      0      0              000010100D    17
 212      1      8      2      0                                    0D    18
 110     11      0      1      0      0      0              000010100D    19
 110      1      5      2      0                                    0D    20
```

Parameter Data Section

The parameter data section of the file contains the parameter data associated with each entity. Parameter data are placed in free format with the first field always containing the entity type number. Therefore, the entity type number and a parameter delimiter precede parameters of each entity. The free format part of a parameter line ends in Column 64. Column 65 contains a blank. Column 66 through 72 on all parameter lines contain the sequence number of the first line in the directory entry of the entity for which parameter data is being presented. Column 73 of all lines in the parameter section shall contain the letter P and Columns 74 through 80 shall contain the sequence number.

For example:

```
314,100.,0.,39.21569,4Hcol8,0;                                          15P      8
212,1,8,61.2541,10.23737,1,1.570796,0.,0,0,-6.923902,-17.59825,          17P      9
```

Terminate Section

The last line in the file is the terminate section, which contains totals of the lines in each of the preceding sections. The terminate section is divided into ten fields of eight columns each, and cannot contain more than one line. The Terminate Section has a “T” in Column 73 and Columns 74 through 80 contain a sequence number with a value of one.

For example:

```
S      1G      4D     38P     29                                                T      1
```

❑

Chapter 9
The Read-IGES Utility

The **Read-IGES** Utility converts files in the IGES format to files in the **Cimatron** part file (**.pfm)** format.

Read-IGES uses a parameter file (**.ipr**). This file contains parameters that control the operation mode and determine the user interaction (refer to the section **Read-IGES Parameters**), as well as optional user filters which provide selective translation of IGES entities (refer to the section **Selective Translation of IGES Files to Cimatron Files**).

For more information about the parameter file and the format of its statements, refer to Chapter 11, **The IGES Utilities Parameter File**.

Translation data can either be displayed on the screen or optionally saved to a log file. The log file contains information such as user parameter values, IGES file global-section data, a listing of each entity translated and any errors or problems encountered during its translation, and statistical information on the entities in the IGES file.

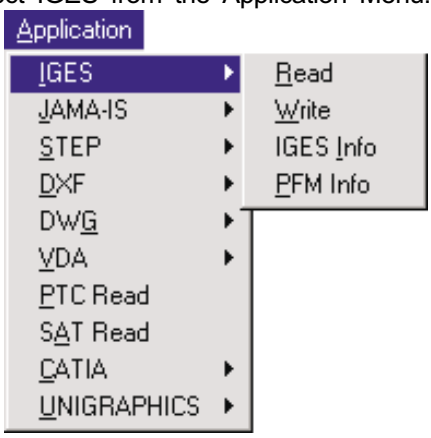
In addition, the IGES file start-section information can optionally be saved to a file.

Running Read-IGES

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select IGES from the Application Menu. The following options are displayed:



3. Select the IGES, Read option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following IGES options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
4  ->  IGES  File  Information  option
5  ->  PFM  File  Information  option
```

3. Select the **Read** option.

Running Modes

Read-IGES may be run interactively or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

*Or*in **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

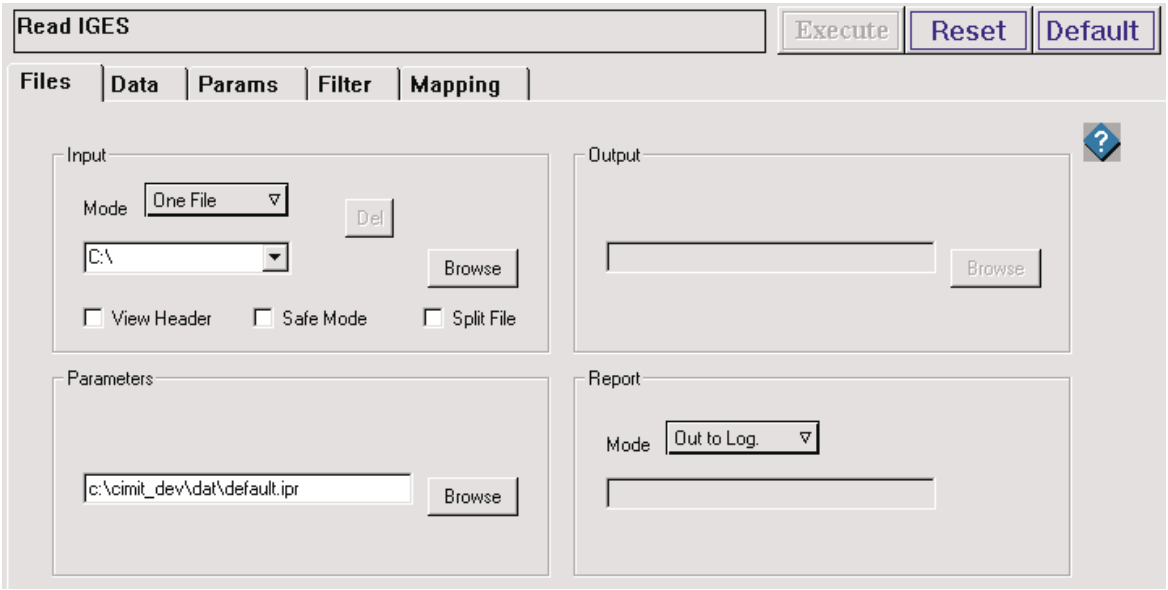
1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file containing one file name per line.

Notes:

- Multi mode is only available when you use the graphic interface for this utility.
- Each part file created will be in the same directory and have the same name as the IGES file, but will have the extension **.pfm** instead of **.igs**.
- Both the IGES file and the generated part file may be located in the current or other directories. Enter the full path name of the file if it is not located in the current directory.
- File extensions may be typed, unless otherwise specified. If the file name is not found as it is typed, the program will search for the same file name with the extension **.igs**.
- In **Batch Mode Without Backup**, if a file already exists with the same name, including the **.pfm** extension, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and **.pfm** files already exist with the name specified in the list, the old **.pfm** file will be copied to a backup file with the extension **.bac**. Then a new **.pfm** file will be created from the IGES file.

Graphic Interaction with Read-IGES



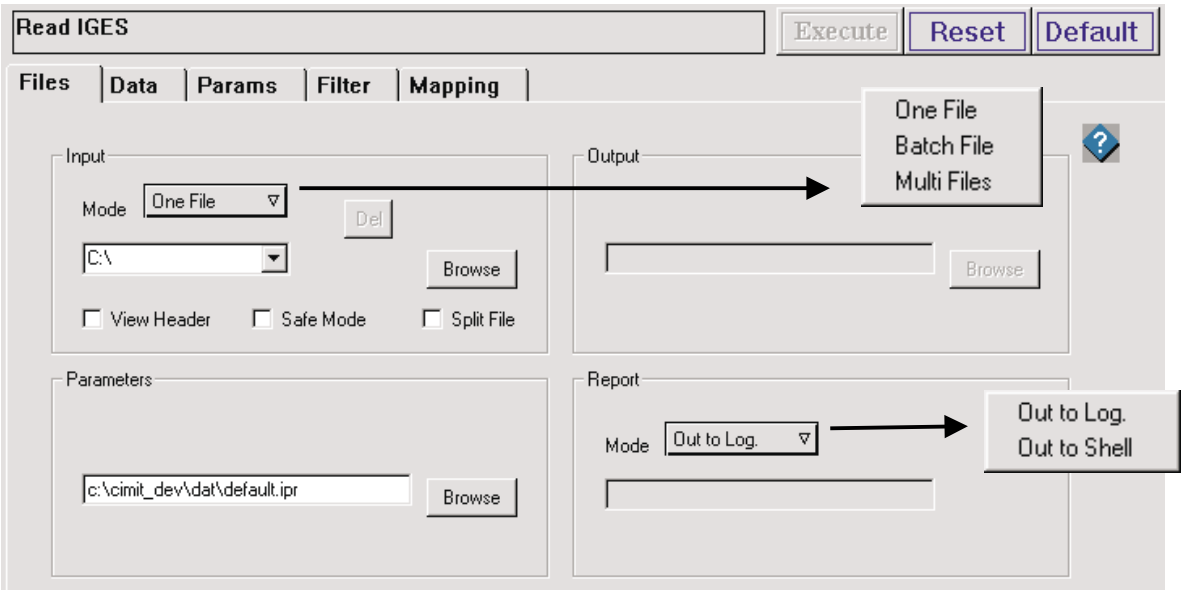
How To:

1. Select the appropriate tab: Files, Data, Param, Filter or Mapping.
2. Set the parameters.
3. Select a file in the Input area using the Browse button.
4. Press the Execute button.

The interaction for each of the tabs is described on the following pages.

Files

This dialog defines the Input, Output, Report and Parameter files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required **Input Mode**: **One File**, **Batch File** or **Multi Files**.
2. Using the Browse button, choose the input **.igs** file (the file to be converted to a Cimatron **.pfm**).
3. Select the appropriate options at the bottom of the Input section: **View Header**, **Safe Mode** and **Split File**.

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the IGES file name from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Select the appropriate options at the bottom of the Input section: **View Header**, **Safe Mode** and **Split File**.

- View Header — When converting to pfm mode, choose whether to display the header information in the converted file.
- Safe Mode — Create a pfm file even if the translation process is interrupted.

  If a failure occurs during the translation process and the application terminates, all the data that was already translated is not saved. When using this option, all the data that was converted up to the failure, is saved.
- Split File — Split to multiple .pfm files, in order to process more than 64,000 entities.

  When you select the **Split File** option, a new directory, named after the input file, is created. The input data is split between several PFM files, which are created in the new directory. The PFM files are named "model_x", where x is the running index of the translation process.

  Another PFM file, named "assembly" is created in the same directory. This file includes each of the "model_x" files as sub-assemblies.

Output

How To:

1. By default, the prefix of the output **.pfm** file is identical to the prefix of the input **.igs** file.

   Choose a different **.pfm** file, or using the Browse button, select an existing **.pfm** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the IGES parameter file is **default.ipr.** You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read IGES translation data **ril** file is identical to the prefix of the input **.igs** file.

   Choose a different **.ril** file, or using the browse button, select an existing **.ril** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

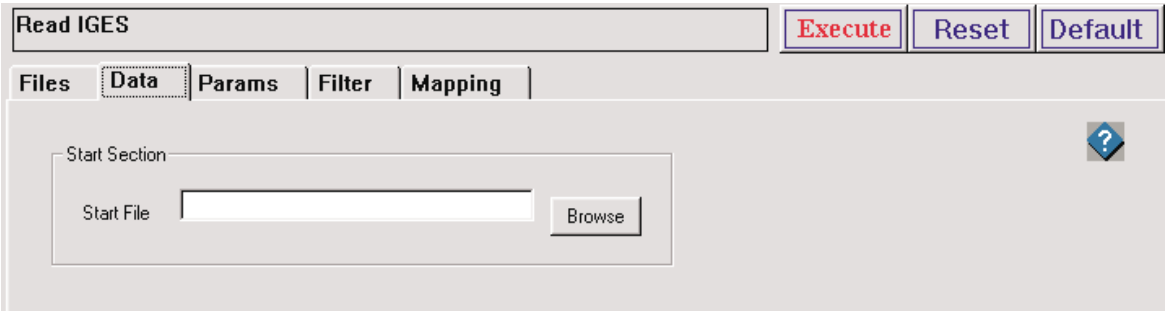
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Data

This dialog is used to specify the name of the file in which start section information from the IGES will be stored (optional).



How To:

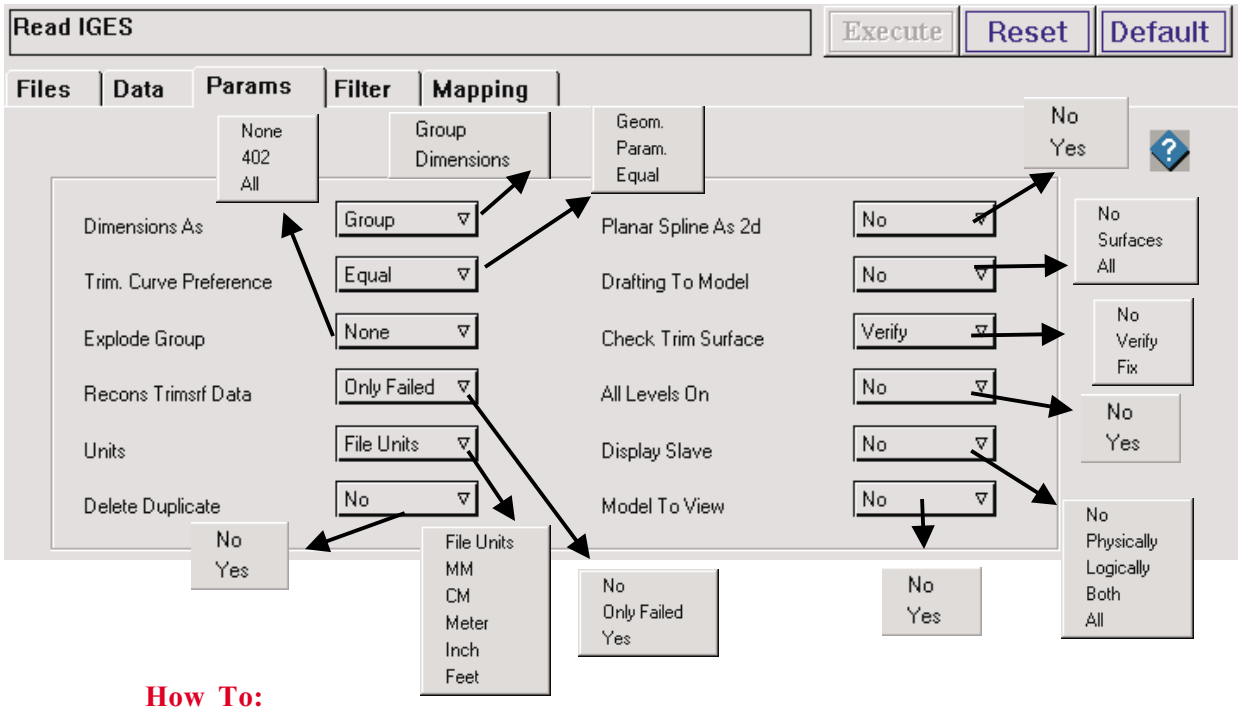
1. Enter the name of the Start File in the Start File field.

Or:

1. Use the Browse button to select an existing .ris file.

Params

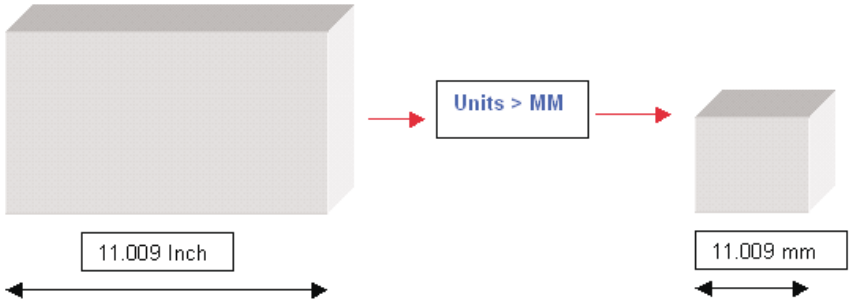
This dialog defines the parameters used during the file translation.



How To:

1. **Dimensions As**: Read IGES dimensions as instances (Group) or as dimensions (Dimensions). See Graphic Interaction description on page 11-10.
2. **Trim. Curve Preference**: Define how trimmed surfaces will be represented (Geom, Param or Equal). See Graphic Interaction description on page 11-10.
3. **Explode Group**: Automatically explode All group entities, entities of type 402, or none.
4. **Recons Trimsrf Data**: Reconstruct trim surface data for all surfaces converted to Cimatron, or only for those surfaces that were not converted correctly, or don't reconstruct trim surface data at all.
5. **Units**: Select the units of the translated file.

   Note: This parameter changes the *unit of measurement* but not the *measurement value*. For example an 11.009 inch part will be converted to 11.009 mm. In other words the *size* of the part will change.

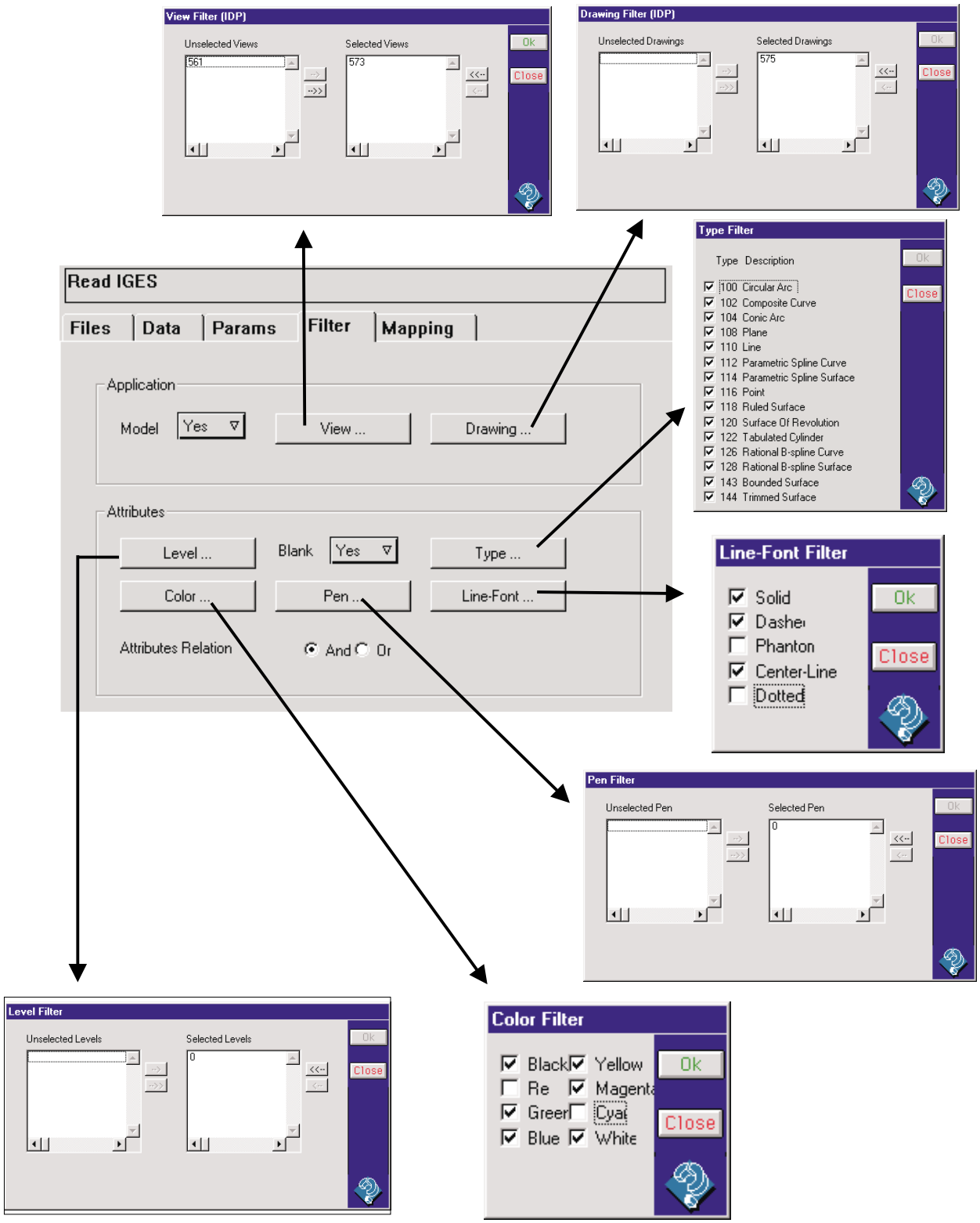


To convert a part to its equivalent size in another unit of measurement, use the **units.dll** User function (**USER >> GENERAL >> units.dll**) after you have translated the IGES file to PFM.

6. **Delete Duplicate**: Delete all the duplicated entities that were created during the translation process.
7. **Planar Spline as 2D**: Convert planar splines to 2D (No or Yes).
8. **Drafting To Model**: Choose whether to translate all geometric entities, only surfaces, or nothing, from IGES views to Cimatron models (No, Surfaces or All).
9. **Check Trim Surfaces**: Mode of fixing trimmed surfaces (No, Verify or Fix).
10. **All Levels On**: Choose which levels are to be displayed after the translation (Yes = display all levels, No = display active level only).
11. **Display Slave**: Choose whether to display all, some or none of the slave entities (No, Physically, Logically, Both or All).
12. **Model To View**: Choose whether to display drafting entities in a model or a dependent view. See description (R-DRAFTING-ENTITIES-TO) on page 11-6.

Filter

This dialog defines the filter parameters used during the file translation.



How To:

Application

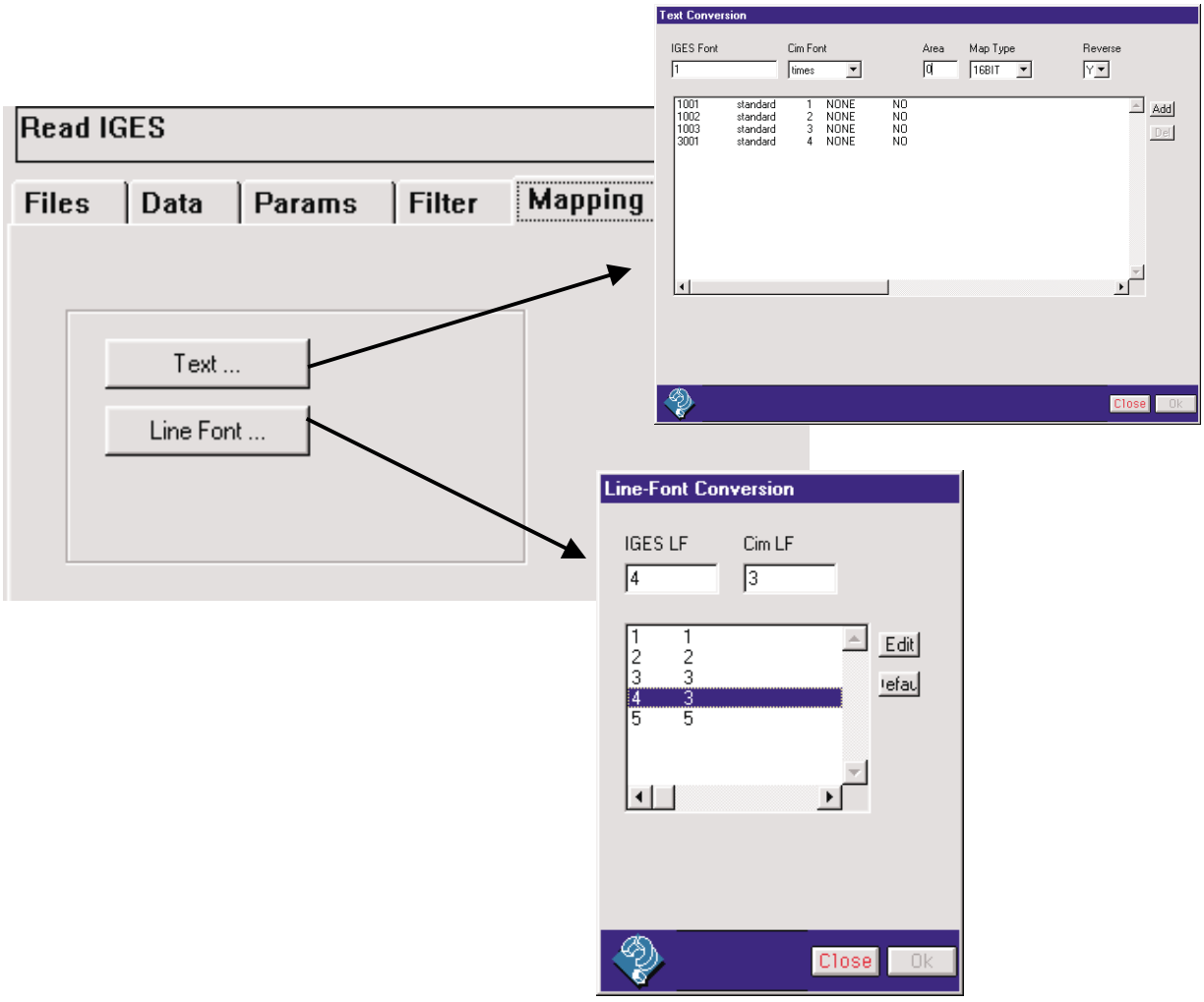
1. Model: Include or exclude the model in the translation.
2. View: Select the views to be included in the translation.
3. Drawing: Select the drawings to be included in the translation.

Attributes

4. Level: Select the levels to be included in the translation.
5. Blank: Include or exclude blank entities in the translation.
6. Type: Choose the type of entity that you want to convert. By default, all the entity types will be converted.
7. Color: Select the colors to be included. All entities in these colors will be included in the translation.
8. Pen: Select the pens to be included. All entities in these pens will be included in the translation.
9. Line Font: Select the line types to be included. All entities with these line types will be included in the translation.
10. Attributes Relation: Select the logical operator to be used - AND/OR.

Mapping

This dialog defines the mapping characteristics for text and line font.



How To:

1. Text: Map the source IGES font with the equivalent Cimatron font.
2. Line Font: Map the source IGES line with the equivalent Cimatron line font.

Textual Interaction with Read-IGES

The following section describes the interaction upon invoking **Read-IGES**. (The interaction that is shown will occur when all the interaction parameters in the parameter file (**.ipr**) are set to ASK.) All prompts that appear on the screen are written here in italics.

IGES (version 5.3) - Cimatron (version 11.0) Convertor <version number>
==========

Enter IGES parameters file name (<CR> = default):

Type name of a parameter file; do not type the **.ipr** extension. To use the default parameter file (**default.ipr**), press <CR>.

Available modes:

The mode menu will appear.

1 -> Interactive mode
2 -> Batch mode with backup
3 -> Batch mode without backup

Enter mode (0 = exit)

Type the number of the appropriate operation mode. To exit the program at this point, type 0 (zero).

If Interactive mode is selected:

Enter IGES file name (<CR> = mode menu) :

Type the name of the IGES file to be converted; do not type the **.igs** extension. To exit the program at this point, press <CR>.

Enter part file name (<CR> = <file_name>) :

Type a name for the part file. To use the name displayed in parentheses, press <CR>.
If the specified file already exists, the following message appears:

—- File <file_name>.pfm already exists
Replace existing file ? (y/n)

To write over the existing file, type **y**. To type a different name for the file, type **n**.

Save IGES log file (Y/N) ? (<CR> = Y) :

You will be asked whether you want to save the **Read-IGES** log file.

If you enter **y**, all translation data will be saved to a log file. The name of the log file will be that of the IGES file with the extension **.ril**.

If you enter **n**, the translation data will be displayed on the screen.

Enter start-section file name(<CR>=<file_name>.ris):

Enter the name of the start-section file. To use the name that appears in parentheses, press <CR>.

Reading IGES file ...

Setting up part file <file_name> ...

The new part file will be saved with the **.pfm** extension.

Entity Mapping

Geometric Entities

Table 9-1: IGES Geometric Entities Supported by the Read-IGES Utility

IGES Entity			Cimatron Entity			Notes
Name	**Type**	**Form**	**Name**	**Class**	**Type**	
Circular Arc	100		Arc	1	2	
Composite Curve	102		2D Composite Curve 3D Composite Curve	1 2	15 15	
Conic Arc	104	All	Conic	1	3	
Copious Data	106	11/63 12/13	2D String 3D String	1 2	5 5	
Plane	108	All	Planar Face	4	1	
Line	110		Line	1	1	
Parametric Spline Curve	112		2D Bezier Spline 3D Bezier Spline	1 2	4 4	
Parametric Spline Surface	114		IGES Surface	3	6	
Point	116		Point	0	1	
Ruled Surface	118	All	Ruled Surface	3	3	(1)
Surface of Revolution	120		Surface of Revolution	3	4	
Tabulated Cylinder	122		Parallel Drive Surface	3	2	
Transformation Matrix	124	0/1/10	Transformation Matrix	15	13	
Rational B-spline Curve	126	All	NURBS (Non-Uniform Rational B-spline) Curve	1	4	
Rational B-spline Surface	128	All	NURBS (Non-Uniform Rational B-spline) Surface	3	11	
Offset Curve	130		Offset Curve	1	7	
Offset Surface	140					(2)
Boundary	141					(3)
Curve on Parametric Surface	142					
Bounded Surface	143		Trimmed Surface	4	2	(4)
Trimmed Surface	144		Trimmed Surface Planar Face	4 4	2 1	(5)

Notes:

- Ruled Surface entity (Type 118, Form 0) is not supported, but is read to **Cimatron** as if its form is 1.
- Offset Surface entity (Type 140) is implemented by creating the base surface and setting its offset value appropriately. Offset to Plane entity (Type 108) is also implemented by adding offset parameter to the depth of the plane.
- Boundary entity (Type 141) is implemented only as part of a Bounded Surface entity (Type 143).
- Bounded Surface entity (Type 143) is translated only as Trimmed Surface in **Cimatron**. If it points to a Plane entity (Type 108), the Bounded Surface entity is not translated to **Cimatron**. Only the Plane entity and the geometric boundary are translated.
- Trimmed Surface entity (Type 144) is translated to a Trimmed Surface or Planar Face in **Cimatron** according to the untrimmed surface type it points to.

Annotation Entities

Table 9-2: IGES Annotation Entities Supported by the Read-IGES Utility

IGES Entity			Cimatron Entity			Notes
Name	**Type**	**Form**	**Name**	**Class**	**Type**	
Copius Data	106	20/21 31-38 40	Group of Lines Hatch Witness Line	13 6	1 2	(1) As part of dimensions
Angular Dimension	202		Group of Lines, Arcs and Notes or Dimension	13 5	1 3	(3)
Diameter Dimension	206		Group of Lines, Arcs and Notes or Dimension	5	2	(3)
General Label	210		Group of Lines and Notes	13	1	
General Note	212	All	Note	6	5	
Leader (Arrow)	214	All	Lines and Arcs	1	1/2	(2)
Linear Dimension	216		Group of Lines, Arcs and Notes or Dimension	13 5	1 1	(3)
Ordinate Dimension	218	All	Group of Lines, Arcs and Notes or Dimension	13 5	1 4	(3)
Point Dimension	220		Group of Lines, Arcs and Notes	13	1	
Radius Dimension	222		Group of Lines, Arcs and Notes or Dimension	13 5	1 2	(3)
General Symbol	228		Group of Geometric Entities and Notes	13	1	
Sectioned Area	230		Hatch	6	2	(1)

Notes:

- All IGES fill patterns of Section entity (Type 106 Forms 31 - 38) and Sectioned Area entity (Type 230) are mapped to the **Cimatron** General fill pattern in a created Hatch entity.

- Arrowhead types defined by the form number of Leader entity (Type 214) are not fully supported. Table 9-3 shows the mapping of IGES arrowhead types to **Cimatron** arrowhead types.

Table 9-3: Mapping of IGES Arrowhead Types to Cimatron Arrowhead Types.

IGES Arrowhead Form	Cimatron Arrowhead Type
1 - Wedge 2 - Triangle 3 - Filled Triangle 7 - Rectangle 8 - Filled Rectangle 11- Open Triangle	Arrow
9 - Slash 10- Integral Sign	Slash
5 - Circle 6 - Filled Circle 12 -Dimension Origin	Circle
4 - No Arrowhead	No Arrowhead

Structure Entities

Table 9-4: IGES Structure Entities Supported by the Read-IGES Utility

IGES Entity			Cimatron Entity			Note
Name	Type	Form	Name	Class	Type	
Null	0					(1)
Subfigure Definition	308		Internal Master	14	1	
Color Definition	314		Color			(2)
Network Subfigure Definition	320		Internal Master	14	1	(3)
Associativity Instance	402	1/7/14/15 3 13/16	Group Visibility Mask	13	1	(4) (5) (6)
Drawing	404		Drawing			
Property	406	1	Level			(7)
Singular Subfigure Instance	408		Instance	13	1	
View	410		View	15	3	
Network Subfigure Instance	420		Instance	13	1	

Notes:

- Null entity (Type 0) is ignored by the Read-IGES utility as defined in the IGES specification. Null entity was defined in the IGES standard to enable editing of the IGES file.
- Color entity (Type 314) is mapped to the closest color definition defined in the **Cimatron** color menu.
- Network Subfigure Definition (Type 320) is created as **Cimatron** Internal Master entity, but only associated geometric entities are included in the Internal Master definition.

- All the Associativity Instance entities (Type 402, Form 1,7,14,15) that are translated to **Cimatron** Group entity are exploded automatically if they include ENTITIES from both modeling and drafting applications.
- Views Visible Associativity Instance (Type 402, Form 3) is translated to the **Cimatron** visibility mask attribute of the entity pointing to it.
- Dimensioned Geometry Associativity (Type 402, Form 13) and Planar Associativity (Type 402, Form 16) are always exploded.
- Definition Levels Property (Type 406, Form 1) is not supported by the Read-IGES utility. When this property is pointed to by an entity, a level is created in **Cimatron**, and its name is the level attribute of the entity. For example, if entity level attribute is the pointer (-24) then a level named “L-0024” is created in **Cimatron**.

❑

Chapter 10
The Write-IGES Utility

The **Write-IGES** utility converts **Cimatron** part files (**.pfm**) to files in the IGES format.

Write-IGES uses a parameter file (**.ipr**). This file contains parameters that control the operation mode and determine the user interaction (refer to the section **Write-IGES Parameters** in this chapter), as well as optional user filters which provide selective translation of IGES entities (refer to the section **Selective Translation of Cimatron Files to IGES Files** in this chapter). For more information about the parameter file and the format of its statements, refer to Chapter 11, **The IGES Utilities Parameter File.**

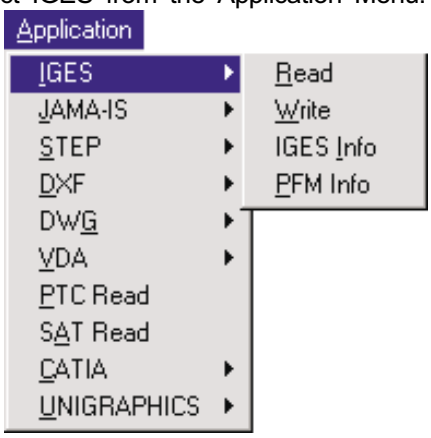
Translation data can either be displayed on the screen or optionally saved to a log file. The log file contains information such as user parameter values, global-section data written to the IGES file, a listing of each entity translated and any errors or problems encountered during translation, and statistical information on the entities contained in the **Cimatron** file.

Running Write-IGES

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select IGES from the Application Menu. The following options are displayed:



3. Select the **IGES, Write** option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following IGES options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
4  ->  IGES  File  Information  option
5  ->  PFM  File  Information  option
```

3. Select the **Write** option.

Running Modes

Write-IGES may be run interactively or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or in **MULTI** mode:

1. Select multiple files.

Batch

In **BATCH** mode:

How To:

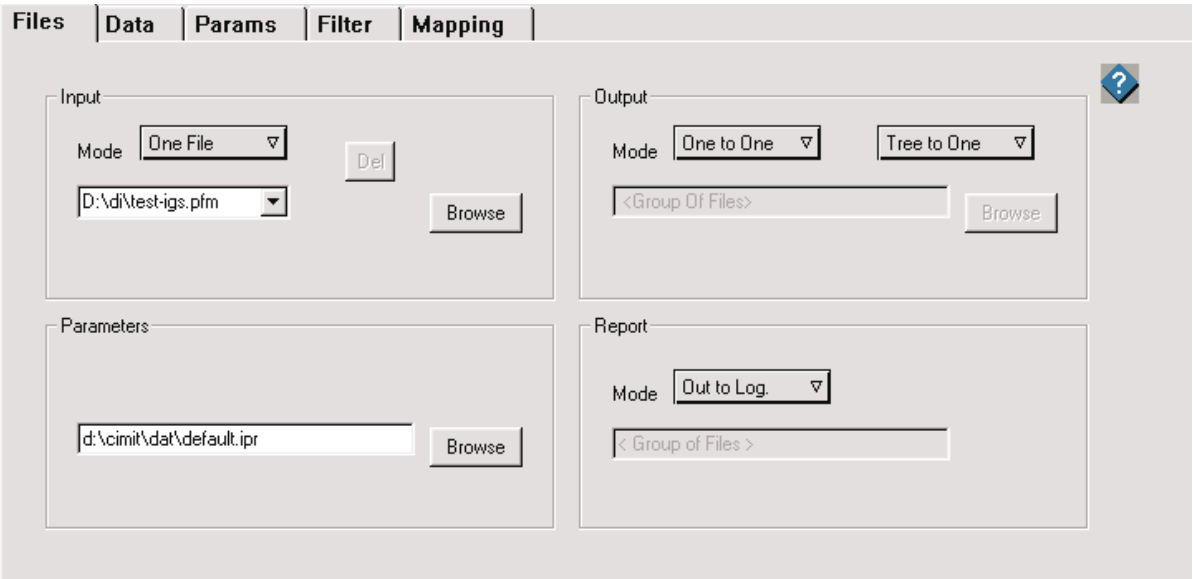
1) Read part file names from a list in a batch file.
2) Convert each file in the list.

In **BATCH** mode a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file containing one file name per line.

Notes:

- Multi mode is only available when you use the graphic interface for this utility.
- Each IGES file created in BATCH mode, or in INTERACTIVE mode if no other name is entered, will be in the same directory and have the same name as the part file, but will have the extension **.igs** instead of **.pfm**.
- Both the part file and the generated IGES file may be located in the current or other directories. Enter the full path name of the file if it is not located in the current directory.
- In BATCH mode, if a file already exists with the same name, including the **.igs** extension, the file created with this program will replace it.

Graphic Interaction with Write-IGES

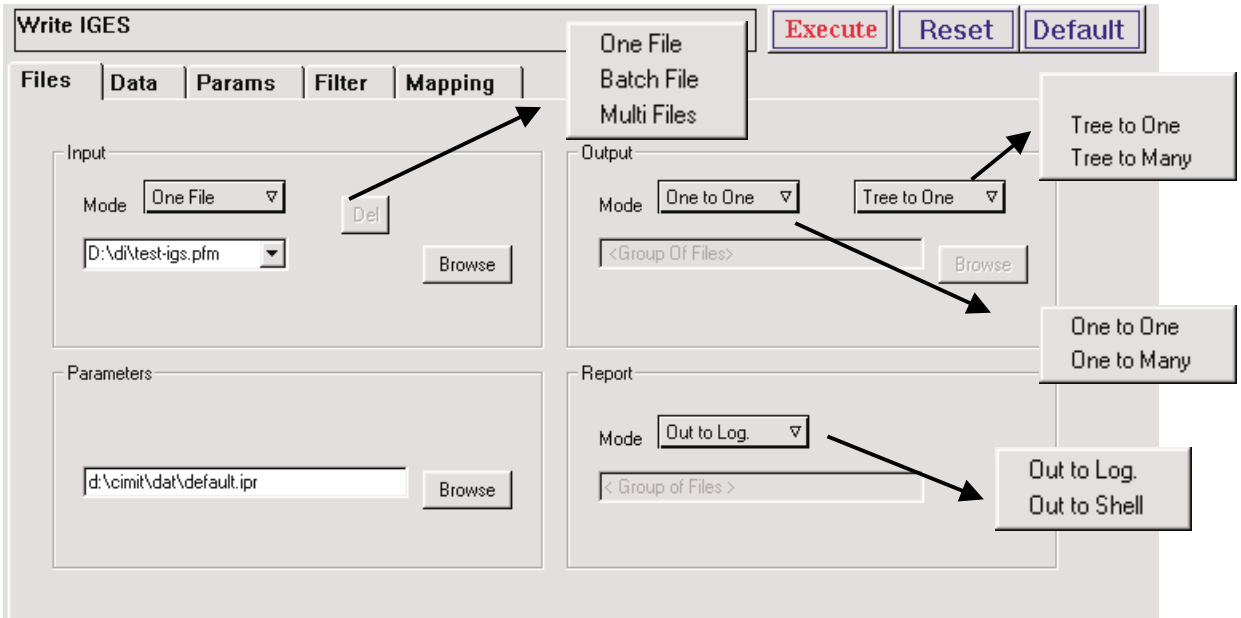


How To:

1. Select the appropriate tab: Files, Data, Params, Filter or Mapping.
2. Set the parameters.
3. Select a file in the Input area using the Browse button.
4. Press the Execute button.

Files

This dialog defines the Input, Output, Report and Parameter files associated with the translation:



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Mode: One File, Batch File or Multi Files.
2. Select the input **.pfm** file (the file to be converted to an IGES file).

The available options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.igs** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.igs** file, or using the Browse button, select an existing **.igs** file. In the latter case, you will be prompted whether or not to overwrite the existing file.
2. Select an Application Filter: ONE TO ONE, ONE TO MANY or ONE TO TWO.
3. Select an Assembly Filter: TREE TO ONE or TREE TO MANY.

The following Application Filters define if an entity is to be translated:

- ONE TO ONE

  All available arguments (all defined views, drawings and model) are included in one file.

  Entities are included according to their visibility in the application. For example, an entity that is visible in an included view will be translated, even if the model in which the entity was created is not included. To give another example, including a drawing causes all views referenced within it to be written to the file.

- ONE TO MANY

  Split the PFM file to many output files. The LOG FILE is not available in this option because many log files are created.

  This option enables you to write the model, each view, or each drawing of a PFM file to a different file.

  The output files are placed in a directory named after the PFM file.

  The view/drawing file name prefix is identical to the translated view/drawing name with an addition (_d for drawing, _v for views).

  The model file name prefix is model.

  If the name of the view or drawing contains special characters, such as a dot (.), dollar sign ($), etc., they will be converted to an underscore (_) in the output file name.

The following Assembly Filters are available:

- TREE TO ONE — Create *one* IGES file for the assembly and drafting models:

  assem_draft.igs.

- TREE TO MANY — Create a directory containing *many* IGES files for the assembly model; one for each part file in the assembly and one IGES file for the drafting model.

  IGES model file names:

  part1.igs, part2.igs, etc.

  IGES drafting file name:

  assem_drf.igs

Parameters

How To:

1. By default, the name of the IGES parameter file is **default.ipr.** You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Write IGES translation data **.wil** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.wil** file, or using the browse button, select an existing **.wil** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

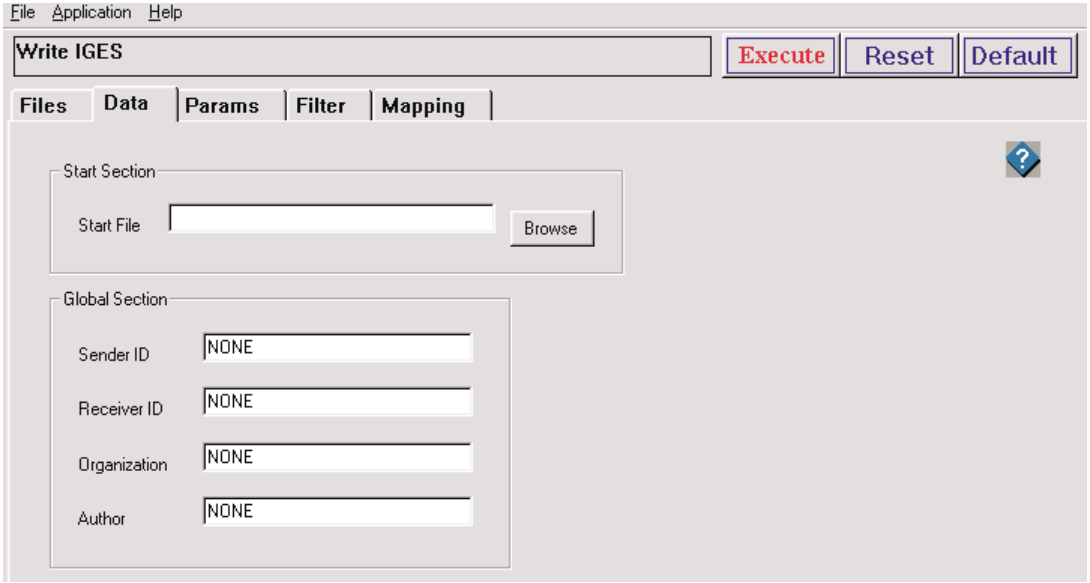
The available options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Data

This dialog is used to specify information about the start section and global section of the IGES file to be written.



How To:

Start Section

1. Enter the name of the Start File in the Start File field.

Or:

1. Use the Browse button to select an existing .wis file.

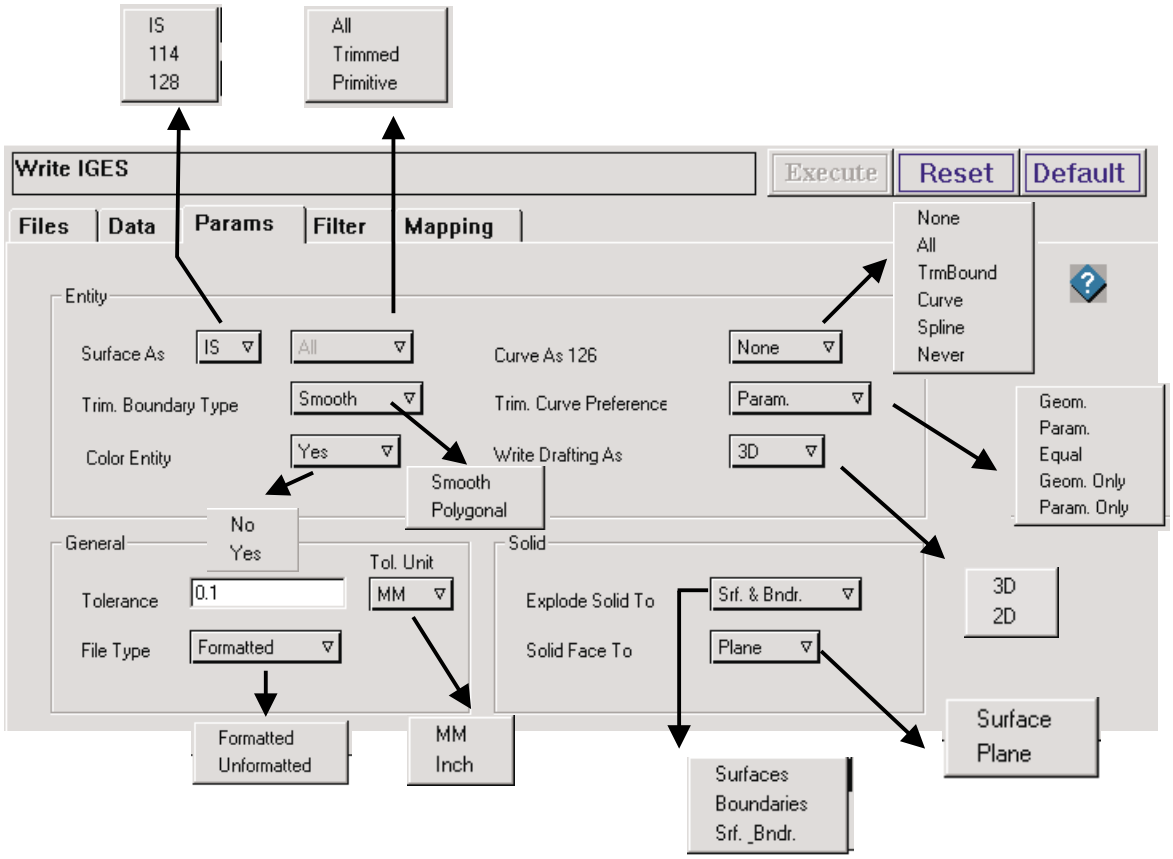
Global Section

Fill in the fields as follows

1. Sender ID: Enter the reference name or identifier used by the sender.
2. Receiver ID: The name used by the receiver to identify the product.
3. Organization: The name of the organization.
4. Author: The name of the author.

Params

This dialog defines the parameters used during the file translation.



How To:

Entity

1. Surface Approximation: Select the method to be used to approximate surfaces as IS, as 114, or as 128, and the type of surfaces to be approximated (All, Trimmed or Primitive).
2. Curve as 126: Type of curve to be approximated to a B-Spline curve.
3. Trim Boundary Type: Define the boundary curve representation: Smooth or Polygonal
4. Trim Curve Preference: Define how the entity Curve on a Parametric Surface is to be represented in the IGES file.
5. Color Entity: Create a color entity or approximate to the predefined IGES entity.
6. Write Drafting As: Write views as 2D or 3D.

General

7. Tolerance: Set the tolerance.
8. Tol. Unit: Set the units for the tolerance in MM or Inches.
9 File Type: Formatted or unformatted.

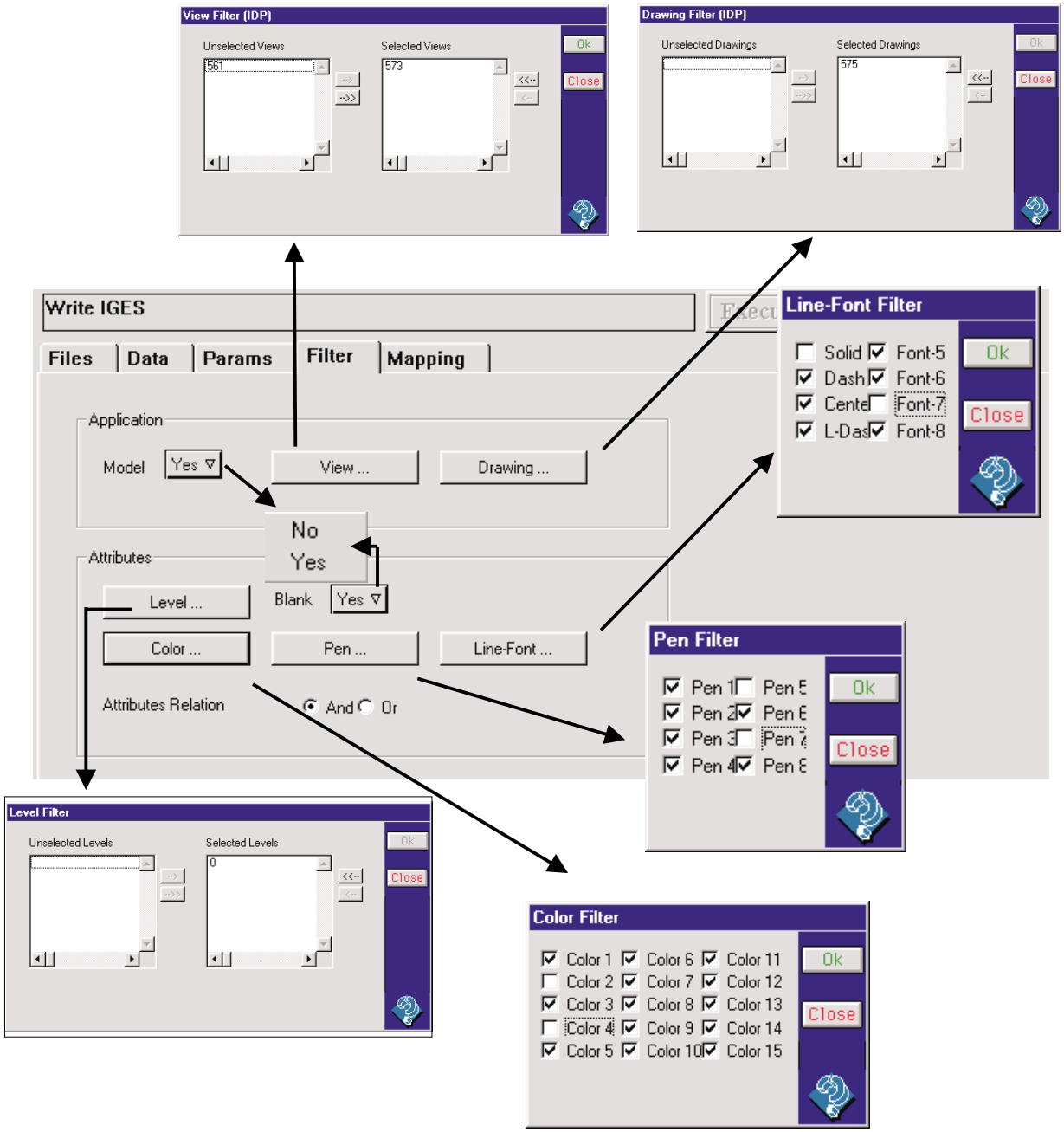
Solid

10. Explode Solid To: Type to which a solid model will be exploded.
11. Solid Face To: Type to which the planar faces will be exploded.

For more details on all the parameters, see Graphical Interaction Description on page 11-21.

Filter

This dialog defines the filters used during the file translation.



How To:

Application

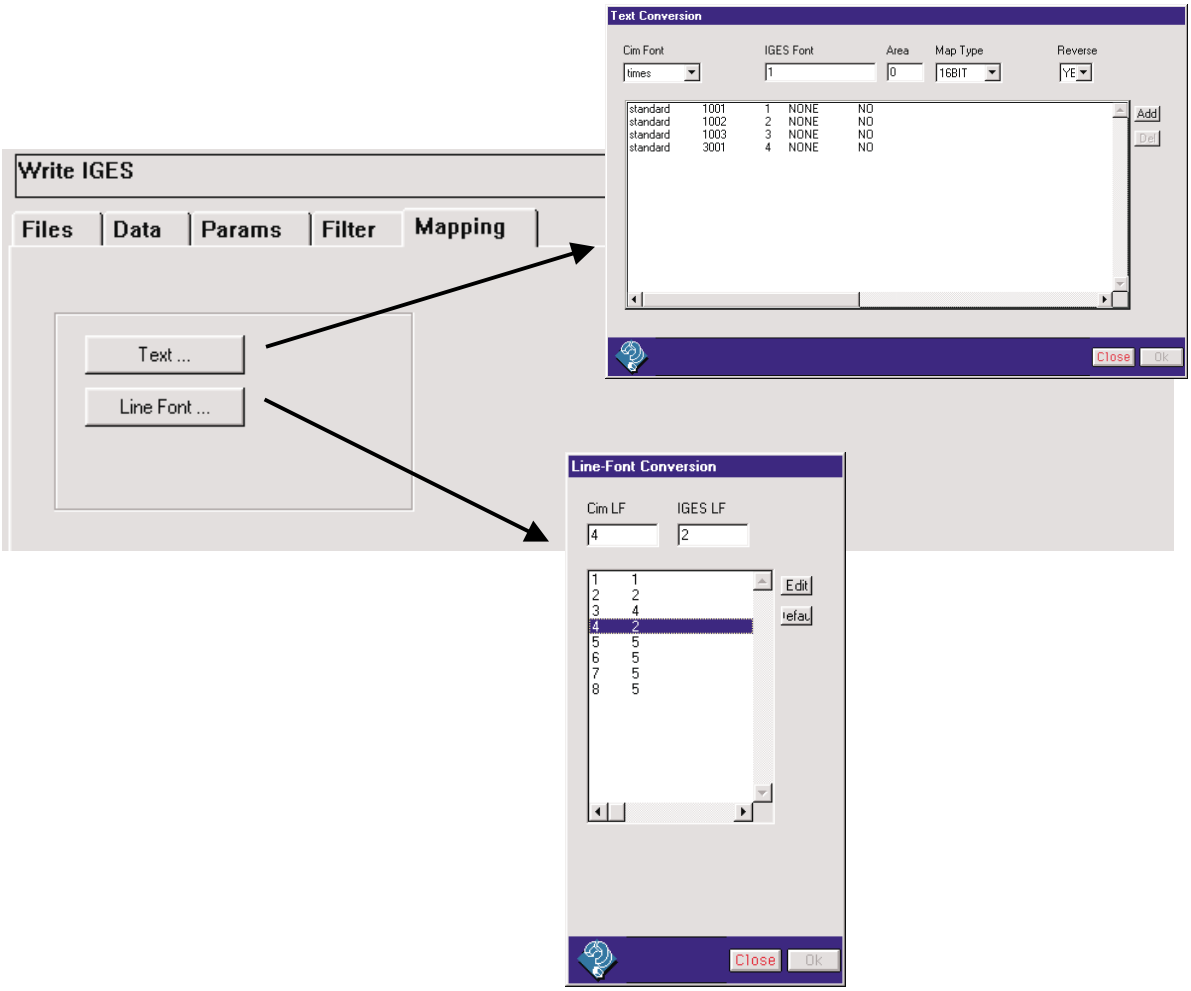
1. Model: Include or exclude the model in the translation.
2. View: Select the views to be included in the translation.
3. Drawing: Select the drawings to be included in the translation.

Attributes

4. Level: Select the levels to be included in the translation.
5. Blank: Include or exclude blank entities in the translation.
6. Color: Select the colors to be included. All entities in these colors will be included in the translation.
7. Pen: Select the pens to be included. All entities in these pens will be included in the translation.
8. Line Font: Check the line types to be included. All entities with these line types will be included in the translation.
9. Attributes Relation: Select the logical operator to be used - AND/OR.

Mapping

This dialog defines the mapping characteristics for Text and Line Font.



How To:

1. Text: Map the source (Cimatron) font with the equivalent destination (IGES) font.
2. Line Font: Convert a Cimatron line font to an equivalent IGES line font.

Textual Interaction with Write-IGES

The following section describes the interaction upon invoking **Write-IGES**. (The interaction that is shown will occur when all interaction parameters in the parameter file (**.ipr**) are set to ASK.) All screen displays are shown here in italics.

Cimatron (version 11.0) IGES (version 5.3) - Convertor <version number>
==========

Enter IGES parameters file name (<CR> = default):	Type the name of the parameter file without the **.ipr** extension. To use the default parameter file, press <CR>.
Available modes:	The mode menu will appear.
1 -> Interactive mode *2 -> Batch mode*	
Enter mode (0 = exit)	Type the number of the appropriate operation mode. To exit the program at this point, type 0 (zero).

If Interactive mode is selected:

Enter part file name (<CR> = mode menu) :	Type the name of the part file without the **.pfm** extension. To return to the mode menu, press <CR>.
Enter IGES file name (<CR> = <file_name>) :	Enter a name for the IGES file to be created. To use the name displayed in parentheses, press <CR>. If the specified file already exists, the following appears:

—- File <file_name>.igs already exists
Replace existing file? (Y/N)

To write over the existing file, type **y**.
To enter a different name for the new file, type **n**.

Save IGES log file (Y/N) ? (<CR> = Y) :	You will be asked whether you want to save the **Write-IGES** log file. If you enter **y**, all translation data will be saved to a log file. The name of the log file will be that of the IGES file with the extension **.wil**. If you enter **n**, the translation data will be displayed on the screen.
Enter start-section file name (<CR>=<file_name>.wis)	Enter the name of the start-section file. To use the name that appears in parentheses, press <CR>.

Setting up part file <file_name>

Entity Mapping

Geometric Entities

Table 10-1: Geometric Entities Supported by the Write-IGES Utility

Cimatron Entity			IGES Entity			Note
Name	**Class**	**Type**	**Name**	**Type**	**Form**	
Point	0	1	Point	116	0	
Line	1	1	Line	110	0	
Arc	1	2	Circular Arc	100	0	
Conic	1	3	Conic	104	0/1	
2D Spline Curve	1	4	Parametric Spline Curve Rational B-spline Curve	112 126	0 0	
2D String	1	5/9	Copious Data	106	11	
Projected Curve	1	6	Parametric Spline Curve	112	0	(1)
Offset Curve	1	7	Offset Curve	130	0	(2)
2D Cubic Spline Curve	1	8	Parametric Spline Curve	112	0	
2D Composite Curve	1	15	Composite Curve	102	0	
Helix	2	2	Parametric Spline Curve	112	0	
3D Spline Curve	2	4	Parametric Spline Curve Rational B-spline Curve	112 126	0 0	
3D String	2	5/6	Copious Data	106	12	
3D Composite Curve	2	15	Composite Curve	102	0	
Parallel Drive Surface	3	2	Tabulated Cylinder Parametric Spline Surface	122 114	0 0	(3) (3),(4),(5)
Ruled Surface	3	3	Ruled Surface Parametric Spline Surface	118 114	1 0	(3) (3),(4),(6)
Surface of Revolution	3	4	Surface of Revolution Parametric Spline Surface	120 114	0 0	(3) (3) (4)
IGES Surface	3	6	Parametric Spline Surface	114	0	(3)
NURBS Surface	3	11	Rational B-spline Surface	128	0	(3)
Other Surfaces	3		Parametric Spline Surface	114	0	(3)
Planar Face	4	1	Plane Entity Single Parent	108 402	0/1/-1 9	
Trimmed Surface	4	2	Trimmed Surface	144	0	

Notes:

- **Cimatron** curve is approximated by IGES Parametric Spline Curve entity (Type 112).
- **Cimatron** Offset curve is approximated by IGES Parametric Spline Curve entity (Type 112) if the base curve is a Projected Curve and/or a 3D curve.

- **Cimatron** surface with offset is written as Offset Surface entity (Type 140).
- **Cimatron** Parallel Drive, Ruled and Revolution surfaces are approximated by IGES Parametric Spline Surface (Type 114) if:

  - constructing curves have been approximated

  - parametric trimming has been performed
- **Cimatron** Parallel Drive Surface with drive curve that is not a line is approximated by IGES Parametric Spline Surface (Type 114).
- **Cimatron** Ruled Surface that is constructed on a Composite Curve or Contour is approximated by IGES Parametric Spline Surface (Type 114).

Annotation Entities

Table 10-2: Annotation Entities Supported by the Write-IGES Utility

Cimatron Entity			IGES Entity			Note
Name	**Class**	**Type**	**Name**	**Type**	**Form**	
Linear Dimension	5	1	Linear Dimension	216	0	
Radial Dimension	5	2	Radius Dimension Diameter Dimension Linear Dimension	222 206 216	0 0 0	(1)
Angular Dimension	5	3	Angular Dimension	202	0	
Ordinate Dimension	5	4	Ordinate Dimension General Symbol	218 228	0 0	(2)
Chamfer Dimension	5	5	General Symbol	228	0	
Arc-Len Dimension	5	6	General Symbol	228	0	
Geometric Tolerance	5	10	General Symbol	228	0	
Section-Line	6	1	General Symbol	228	0	
Hatch	6	2	Copious Data	106	31- 38	
Note	6	5	General Note	212	0	
ID Num	6	7	General Symbol	228	0	
B.O.M.	6	9	General Symbol	228	0	
Label	6	4/10 4	General Symbol General Label	228 210	0 0	(3)

Notes:

- **Cimatron** Radial Dimension is written as IGES Linear Dimension entity (Type 216) when Witness Lines (Type 106, Form 40) are included in the definition of the dimension.
- **Cimatron** Ordinate Dimension is written as IGES General Symbol entity (Type 228) when Leader entities (Type 214) are included in the definition of the dimension.
- **Cimatron** Old Label (Class 6, Type 4) is written as IGES General Label entity (Type 210) when no geometric elements are included in the label definitions.

Structure Entities

Table 16-3: Structure Entities Supported by the Write-IGES Utility

Cimatron Entity			IGES Entity			Note
Name	**Class**	**Type**	**Name**	**Type**	**Form**	
Instance	13	1	Subfigure Instance	408	0	
Internal Master	14	1	Subfigure Definition	308	0	
Group	14	9	Associativity Instance	402	1	
View			View	410	0	
Drawing			Drawing	404	0	(1)

Note:

- **Cimatron** Views located inside a Drawing with a rotation angle are written to IGES Drawing entity (Type 404) without the rotation.

Constraints

- Slave entities are never written by themselves, but as part of their parent entity, i.e. only independent parent entities can be filtered or unfiltered.
- Master entities are always written. As a result, a level that is excluded by the user may be written to the IGES file because of Master entities that are located on the level.
- Instances and Groups are written according to application group filters only. Their inclusion is not affected by their attributes.
- There are limits on the number of argument values accepted for some attribute filters, as shown in Table 10-5.

Table 10-5. Maximum Number of Argument Values for Attribute Filter Arguments

Filter Argument	Maximum No. of Values
DEPENDENT VIEW	47
LEVEL	500
COLOR	15
LINE-FONT	8
PEN	8
SET	1

❑

Chapter 11
The IGES Utilities Parameter File

The IGES Utilities parameter file allows the IGES utilities user to:

- Customize the interaction with the utilities.
- Insert data required by the utilities.
- Influence the behavior of the utilities by means of user-filters.

This chapter contains a general description of the creation, format and use of the parameter file. Further information on the **Read-IGES** parameters can be found in Chapter 9, **The Read-IGES Utility.** Further information on the **Write-IGES** parameters can be found in Chapter 10, **The Write-IGES Utility**.

A description of the **Parameter File Check** Utility is given at the end of this chapter.

Parameter File Name

The format of the parameter file name is **filename.ipr**. There are no limitations on the name, however the extension must be **.ipr**. For example, the name of the default parameter file is **default.ipr**.

Parameter File Creation

The **Cimatron** IGES parameter file is created by the user. Since the file is in ASCII (American Standard Code for Information Interchange) format, it can be easily created using any text editor. The **Cimatron** original default parameter file (**default.ipr),** located in the **<root_cad>\dat** directory, can be used as a template and guide for the user.

Parameter File Format and Rules

The parameter file is an ASCII file. The file contains statements which consist of predefined keywords and their parameter values. Following are some general rules of the parameter file format:

- The order of the lines is not important.
- Letter case is not important except when specifying **Cimatron** files, in which case the **Cimatron** file naming rules must be used.
- Each line starts with a keyword.
- If the same keyword appears more than once, and the statements can be accumulated, they will; otherwise the last statement encountered will be used.
- A line that starts with ## is a comment line.
- Empty lines are ignored.
- Maximum of 50 words are allowed in one statement.
- To continue the arguments of a statement onto an additional line, finish the current line with \<CR> (backslash followed by carriage return) and continue on to the next line.
- The maximum number of characters allowed in one statement is 256.

Parameter File Statements and Syntax

All statements in the parameter file have the following format:

KEYWORD1 KEYWORD2 statement-argument(s) statement-switch(es)

In general, statement arguments are required while statement switches are optional. Statements are divided into three groups according to their function. The first keyword of each statement identifies which group it belongs to:

- SELECT statements are used to select one of a number of predefined options. SELECT statements usually require only one argument. Additional arguments or switches are optional. If an optional argument is not supplied by the user, a default program value is used.
- SET statements provide specific information that is required by a utility.
- MAP statements are used to map values between two sets of data.

Following are the conventions used to describe the parameter file statements:

- | - separates different options of the same argument.
- {A} - argument is required.
- [A] - argument is optional.
- <A> - argument is the default value if no value is given.
- (n-m) - range of possible values.

- Upper case words are fixed predefined keywords or arguments values. They should be typed in the parameter file exactly as they appear here.
- Lower case words are used for variables for which the user would normally substitute a value.
- The keyword ASK means that the user will be prompted to supply information when running the program.

Read-IGES Parameters

The parameter file (**.ipr**) contains parameters that control the operation mode, determine the interaction when running **Read-IGES** and provide information needed by the utility. These parameters will be described here.

The user filter parameters are described separately in the section **Selective Translation of Cimatron Files to IGES Files** on page 11-24.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP |<NOBACKUP>]

  The parameter R-MODE selects the mode of operation.

  Read-IGES may be run interactively or in batch mode. Optional backup is available with batch mode.

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. User is prompted for name of each IGES file to be converted.
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the IGES files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.
BACKUP	Option used in batch mode. If a **.pfm** file already exists with the name specified in the batch file, it will be copied to a backup file with the extension **.bac.**
NOBACKUP	This is a system default option used in batch mode. If a part file (**.pfm**) with the name specified in the list already exists the new part file will replace it.

  Defaults: ASK, NOBACKUP

- SELECT R-VIEW-HEADER {<OFF> | ON}

  When converting to .pfm mode, choose whether to display the header information in the converted file.

  OFF Do not display the header.

  YES Display the header.

  Default: OFF

- SELECT R-SAFE-MODE {<OFF> | ON}

  Create a .pfm file even if the translation process is interrupted.

  If a failure occurs during the translation process and the application terminates, all the data that was already translated is not saved. When using this option, all the data that was converted up to the failure, is saved.

  OFF Do not use the safe mode.

  YES Use the safe mode.

  Default: OFF

- SELECT R-SPLIT-FILE {<OFF> | ON}

  Split to multiple .pfm files, in order to process more than 64,000 entities.

  When you select the **Split File** option, a new directory, named after the input file, is created. The input data is split between several PFM files, which are created in the new directory. The PFM files are named “model_x”, where x is the running index of the translation process.

  Another PFM file, named “assembly” is created in the same directory. This file includes each of the “model_x” files as sub-assemblies.

  OFF Do not split the files.

  YES Split the files.

  Default: OFF

- SELECT R-LOG-FILE {YES | NO | <ASK>}

  The parameter R-LOG-FILE selects log file mode.

  Translation process data can either be saved to a log file or displayed on the screen.

  YES All translation data will be saved to a log file. The name of the log file will be that of the part file with the extension **.ril**. The screen will display only information about the IGES type and form contained in the IGES file.

  NO Translation data will be displayed on the screen.

  ASK User is asked whether he wants to save translation data to log file.

  Default: ASK

- SELECT R-START-SECTION {<NO>|ASK|YES} [start-section-file <iges-file-name.ris>]

  The parameter R-START-SECTION selects the start section information mode.

  The start section information from the IGES file can optionally be written to a separate file. For more information on the start-section, refer to the section **Overview of IGES File Format** in Chapter 8.

  NO A start section file will not be created.

  ASK The user will be prompted for the name of the start-section file. The file will be given the extension **.ris**.

  YES Start section information from the IGES file will be stored in the file specified in the statement. If no file name is specified, the file **<Iges-file-name>.ris** will be used.

  Default: NO

- SELECT R-DIMENSION-AS {<GROUP> | DIMENSION}

  Read dimension as group or as manual dimension.

  GROUP Read dimension from IGES file as Group of lines, Arcs and Notes to **Cimatron**.

  Dimension Read dimension from the IGES file as dimension entity to **Cimatron**.

  Default: GROUP

- SELECT R-BOUNDARY-PREF-TYPE {GEOM | PARAM | <EQUAL>}

  Preference for trimmed Surface Boundary Representation.

  GEOM Read the original surface, represent the boundary through a geometric projection of each point on the boundary onto the original surface.

  The points do not have any relationship with one another and an error, during the read operation, with this option might result in small waves surrounding one, or a group of points.

  PARAM Read the original surface, represent the boundary with a parametric equation which defines the curve, and then trim the original surface.

  If a file is incorrectly read with this option, it is completely unusable - a small mistake in the equation during writing, reading, translating or trimming will propagate through the equation with increasing and accumulating errors.

  However, the PARAM method contains and retains more information from the original file than the GEOM method and should result in better and smoother boundaries.

  EQUAL Read the surface and represent the boundary using the method used to create the original IGES file.

  Default: EQUAL

- SELECT R-EXPLODE-GROUP {<NONE> | 402 | ALL}

  Automatically explode All group entities, entities of type 402, or none.

  Default: NONE

- SELECT R-PLANAR-SPL-AS-2D {<NO> | YES}

  The parameter R-PLANAR-SPL-AS-2D enables you to define a 3D planar spline representation as 2D or not.

  Default: NO

- SELECT R-DRAFTING-TO-MODEL {<NO> | SURFACES | ALL}

  Choose whether to translate all geometric entities, only surfaces, or nothing, from IGES views to Cimatron models (No, Surfaces or All).

  Default: NO

- SELECT R-FIX-TRMSRF {NO| <VERIFY> | FIX}

  Mode of fixing trimmed surfaces.

  Default: VERIFY

- SELECT R-CHECK-WEIGHTS {YES | NO}

  Check the value of the weights in the Nurbs surface header.

  Default: NO

- SELECT R-PEN-WIDTH-SET {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | <FALSE>}

  Set pen width of all entities in read as any value between 1-8, or FALSE.

  Default: FALSE

- SET R_PREDEFINED-LEVELS-NUM {num_levels <0>}

  Keep track of empty levels.

  Default: 0

- SELECT R-RECON-TRS-DATA {NO | <ONLY_FAILED> | YES}

  Reconstruct trim surface data..

NO	The original surface and trimming curves are not displayed.
ONLY_FAILED	The original surface and trimming curves are displayed only for those surfaces that are not converted to Cimatron correctly.
YES	The original surface and trimming curves are displayed for all trimmed surfaces. Using this value may increase the size of the pfm file.

  Default: ONLY_FAILED

- SELECT R-ALL-LEVELS-ON {<NO> | YES}

  Select which level(s) will be displayed after a read IGES operation.

  NO Only the active level will be displayed.

  YES All levels will be displayed.

  Default: NO

- SELECT R-SAFE_MODE {<OFF> | ON}

  Create a PFM file even if the translation process was not completed.

  If a failure occurs during the translation process and the application terminates, all the data that was already translated is not saved. When using this option, all the data that was converted up to the failure, is saved.

  This option causes the translation process to extend extremely.

  OFF Do not use the safe mode.

  ON Use the safe mode.

  Default: OFF

- SELECT R-DRAFTING-ENTITIES-TO {<NO> | YES}

  Choose whether to display drafting entities in a model or a dependent view.

  NO Drafting entities will be created in model.

  YES All drafting entities from the IGES model will be created in a dependent view, called “MDL” (or MDLx, where x is the first free index). If there isn’t a free dependent view (all 47 are occupied), an independent view will be created instead.

  Default: NO

- SELECT R-DELDUP-MODE {<NO> | YES}

  Delete all the duplicated entities that were created during the translation process.

  When using the surface to model parameter, one surface can be duplicated and be placed in several levels. To avoid redundancy the deldup parameter can be used.

  NO Do not use deldup option.

  YES Use the deldup option.

  Default: NO

- SELECT R-SPLIT-SRF {<NO> | YES}

  Split a surface on each point when the surface is not smooth.

  This option checks every surface for smoothness. If a break point is found, the surface is split at the break point and two surfaces created as a result.

  NO Do not split on break points.

  YES Split surfaces on break points.

  Default: NO

- SELECT R-DISP-SLAVE {<NO> | PHYSICALLY | LOGICALLY | BOTH | ALL}

  Display slave entities. This parameter gives you the option to display all or some of the slave entities.

NO	Do not display slave entities. This is the default option.
PHYSICALLY	Display entities that are marked as 'physically dependant'.
LOGICALLY	Display entities that are marked as 'logically dependant'.
BOTH	Display entities that are marked as 'both'.
ALL	Display all the entities.

  Default: NO

- SELECT R-UNITS {<FILE-UNITS> | MM | CM | METER | INCH | FEET}

  Select the units of the translated file.

FILE-UNITS	Select the default file units.
MM	Select MM.
CM	Select CM.
METER	Select METER.
INCH	Select INCH.
FEET	Select FEET.

  Default: FILE-UNITS

- SELECT R-REFINE-TRMSRF-TOL {approximation-tolerance(0.0-1.0) | <0.0>}

  Refine trimmed surface boundaries when required (TOL > 0 and TOL < 1). The default option (0) parameter will not be used in the data interface.

  If the value is between 0 and 1, this will be used as the tolerance for the Refine Trmsrf algorithm which will be forced on the resulting trimmed surfaces after the data interface process is completed.

  If you set the value <0 or >1, the application will assume that 0.0 was used and the parameter will be ignored.

  Default: 0.0

Statements for Text Conversion

- MAP TEXT {iges-font} {cim-font} [{char-map-area-number}]
  MAP R-TEXT {iges-font} {cim-font} [{char-map-area-number}]

  Maps the source (IGES/**Cimatron**) font with the equivalent destination (IGES/**Cimatron**) font. For Read-IGES, this maps the IGES font with the equivalent **Cimatron** font. Characters from the source font are mapped to destination font characters using the character mapping file (**.cdt** file) and the char-map-area-number. Character mapping files are located in the directory **<root_cad>\dat\fonts**. The char-map-area-number is a hexadecimal number indicating the range of source font character codes to be used from the character mapping file when the given source font characters are mapped. Each char-map-area-number specifies a range of 256 characters. The **.cdt** files used by IGES Utilities are:

  riges.cdt - for the Read-IGES option

iges-font	IGES Font Code number of the IGES Font
cim-font	**Cimatron** Font name, one of the fonts located in the directory **<root_cad>\dat\fonts.**
char-map-area-number	Area number (Hexadecimal number) in the character mapping file (**.cdt** file).

  Defaults:

  If the MAP TEXT statement for a specific font is omitted:

  Default **Cimatron** font for Read-IGES : standard

  Default IGES font for Write-IGES : 1

  Default char-map-area-number if omitted : 0

  Other explicit defaults:

  MAP TEXT 1001 standard 1 - for IGES Symbol Font 1 *

  MAP TEXT 1002 standard 2 - for IGES Symbol Font 2 *

  MAP TEXT 1003 standard 3 - for IGES Drafting Font *

  MAP TEXT 3001 standard 4 - for IGES Latin-1 Alphabet Font (ISO8859-1)

Statement for Line-Font Conversion

- MAP R-LINE-FONT {iges-line-font(1-5)} {cim90-line-font(1-8)}

  Map IGES Line-Font to equivalent **Cimatron** Line-Font

  Presence of both line-font parameter values are necessary.

  Defaults:

  0 - IGES No pattern specified - 0 for **Cimatron** line font Solid 1
  1 - IGES Solid line font - 1 for **Cimatron** line font Solid 1
  2 - IGES Dashed line font - 2 for **Cimatron** line font Dash 2
  3 - IGES Phantom line font - 3 for **Cimatron** line font Center 3
  4 - IGES Centerline line font - 4 for **Cimatron** line font Center 3
  5 - IGES Dotted line font - 5 for **Cimatron** line font Font 5

  IGES Line Font Definition entity (Type 304), is translated to **Cimatron** line font Solid 1.

Read-IGES Utility Interaction Statements

Select mode of operation:

- SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP | <NOBACKUP>]

Select log file mode:

- SELECT R-LOG-FILE {YES | NO | <ASK>}

Select start-section information mode:

- SELECT R-START-SECTION {<NO> | ASK | YES} [start-section-file <iges-file-name.ris>]

Additional Read-IGES Utility Statements

Set special text with the equivalent **Cimatron** character-set value:

- SET R-SPECIAL-TEXT {<-1> | cim90-cset-number(0-14)}

- SELECT R-DIMENSION-AS {<GROUP> | DIMENSIONS}

- Preferred type of CURVE-ON-A-PARAMETRIC-SURFACE/BOUNDARY ENTITY (TYP 142/141)

- SELECT R-BOUNDARY-PREF-TYPE {GEOM | PARAM | <EQUAL>}

- SELECT R-EXPLODE-GROUP {<NONE> | 402 | ALL}

- SELECT R-PLANAR-SPL-AS-2D {<NO> | YES}

Write-IGES Parameters

The parameter file (**.ipr)** contains parameters that control the operation mode, determine the interaction when running **Write-IGES**, and provide information needed by the utility. These parameters will be described here.

The user filter parameters are described separately in the section **Selective Translation of Cimatron Files to IGES Files** in this chapter. For more information on the parameter file, see Chapter 11, **The IGES Utilities Parameter File**.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT W-MODE {<ASK> | INTER | BATCH}

  The parameter W-MODE selects the mode of operation.

  Write-IGES may be run interactively (INTER) or in batch mode (BATCH).

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. The user is prompted for the name of each **Cimatron** part file to be converted.
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.

  Default: ASK

- SELECT W-LOG-FILE {YES | NO | <ASK>}

  The parameter W-LOG-FILE selects log file mode.

  Translation process data can either be saved to a log file or displayed on the screen.

YES	All translation data will be saved to a log file. The name of the log file will be that of the IGES file with the extension **.wil**. The screen will display only information about the class and type contained in the part file.
NO	Translation data will be displayed on the screen.
ASK	User is asked whether he wants to save translation data to log file.

  Default: ASK

- SELECT W-START-SECTION {<NO>| ASK | YES} [start-section-file <Iges-file-name.wis]

  The parameter W-START-SECTION selects the start section information mode.

  The start section information for the IGES file can contain either the default text, or text taken from a start-section file.

  The start-section file is an ASCII free format file created by the user. A line length of more than 73 characters is not recommended. For more information, refer to the section **Overview of the IGES File Format** in Chapter 8.

  NO The default text, **Cimatron** IGES convertor, will be written in the start section of the IGES file.

  ASK Start section information will be taken from a file. The user will be prompted for the name of the start-section file.

  YES Start section information will be taken from the file specified in the statement. If no file name is specified, the file **<Iges-file-name>.wis** will be used.

  Default: NO

Additional Statements

- SELECT W-FILE-TYPE {<FORMATTED> | UNFORMATTED}

  The parameter W-FILE-TYPE sets the type of IGES file generated.

  FORMATTED Create a formatted file in which every 80 characters (i.e. every record) is written on a separate line.

  UNFORMATTED Create an unformatted file.

  Default: FORMATTED

- SELECT W-BOUNDARY-PREF-TYPE {GEOM | <PARAM> | EQUAL | GEOM-ONLY | PARAM-ONLY}

  The parameter W-142-PREF-TYPE sets the preferred representation of Curve On a Parametric Surface (Type 142) to be written to the IGES file.

  GEOM Write the surface, representing the boundary through a geometric projection of each point on the boundary onto the original surface.

  The points do not have any relationship with one another and an error, during the write operation, with this option might result in small waves surrounding one, or a group of points.

  PARAM Write the surface, representing the boundary with a parametric equation which defines the curve, and then trim the original surface.

  If a file is incorrectly written with this option, it is completely unusable - a small mistake in the equation during writing, reading, translating or trimming will propagate through the equation with increasing and accumulating errors.

  However, the PARAM method contains and retains more information from the original file than the GEOM method and should result in better and smoother boundaries.

  EQUAL Write the surface and represent the boundary using the method used to create the original IGES file.

  PARAM-ONLY Writes only the parametric representation and sets the flag to parametric preferred.

  GEOM-ONLY Writes only the geometric representation and sets the flag to geometric preferred.

- SET W-TOL {approximation-tolerance (0.001-100) <0.1>}

  The parameter W-TOL sets the tolerance in mm used when approximating a **Cimatron** entity by an IGES entity.

  approximation-tolerance User-defined tolerance; the range is 0.001 - 100 mm.

  Default: 0.1

- SELECT W-BOUNDARY-CURVE-TYPE {<SMOOTH> | POLYGON}

  Preferred boundary curve representation

  Smooth Curve, both in the geometric or parametric representation written as curves.

  Polygon Curve, both in the geometric or parametric representation written as a polygon

  Default: SMOOTH

- SELECT W-3D-VIEW-AS { 2D | <3D> }

  Write Drafting As: Write views as 2D or 3D.

  Default: 3D

- SELECT W-CURVE-AS-126 {<NONE> | ALL | TRIMBOUND. | CURVE | SPLINE}

  Approximate curves to B-spline Curve (Type 126).

  None No curves except trim surface boundaries.

  All All curves.

  Trm.bound. All trimmed boundaries.

  Curve Only curves.

  Spline Only splines.

  Never No curves.

  Default: None.

- SELECT W-SURFACE-AS-114 {PRIMITIVE | TRIMMED | ALL | <NONE>}

  Approximate surfaces to Spline Surface (Type 114).

  Primitive Drive, Ruled, Surfaces of Revolution.

  Trimmed All trimmed surfaces.

  All All surfaces.

  None No surfaces.

- SELECT W-SURFACE-AS-128 { ALL | TRIMMED | PRIMITIVE | <NONE>}

  Approximate surface to Rational B-Spline Surface (Type 128)

  All All surfaces.

  Trimmed All trimmed surfaces.

  Primitive Drive, Ruled, Surfaces of revolution.

  None No surface

  Default: None.

  Notes:

  - The **Cimatron** smooth surface is approximated by the IGES Parametric Rational B-Spline Surface (Type 128), or by IGES Parametric Spline surface (Type 114).
  - Only the last statement is used by the application. If none of the statements (type 114 or type 128) is used, all the surfaces are translated as is.

- SELECT W-TOL-UNITS { <MM> | INCH}

  Select the unit of tolerance in the tolerance field.

  MM The tolerance is in millimeters.

  INCHES The tolerance is in inches.

  Default: MM.

- SELECT W-SOLID-EXPLODE-AS {SURFACES | BOUNDARY | <BOTH>}

  Explode SOLID model to surfaces, boundaries, or both.

  Default: BOTH

  The function will be able to automatically divide the output to two different IGES files: one for the SOLID model and the other for the SOLID drafting application, i.e. VIEWS, SECTION, and DRAWINGS.

  The directory, named according to the user input, contains IGES files. The model is written to a file named **model.igs**. The drafting applications are written to a file named **draft.igs**. If the SPLIT option was selected by the user then drafting applications are written to different files named following the current SPLIT rules. A log-file is created for each IGES file. The log file will include a remark that the file came from the SOLID pfm.

- SELECT W-SOLID-PLANAR-FACE-AS {SURFACE | <FACE>}

  Explosion of SOLID model's planar-faces to planar-faces or planar surfaces.

  Default: FACES

- SELECT W-COLOR-ENTITY {NO | <YES>}

  For colors in Cimatron[it] not predefined in IGES, create as color entity, or approximate to IGES predefined color.

  Default: YES

- SELECT W-DIMENSION-ASSOCIATIVITY {<NO> | YES}

  Create dimensioned geometry associativity or not.

  Default: NO

- SET W-SENDER-PRODUCT-ID {user-string <NONE> }

  The parameter W-SENDER-PRODUCT-ID supplies data needed for the global section of the IGES file.

user-string	The name assigned by the sender to the product represented by the file.
NONE	If no other information is given, the string NONE will be written.

  Default: NONE

- SET W-RECEIVER-PRODUCT-ID {ASK |user-string <NONE>}

  The parameter W-RECEIVER-PRODUCT-ID supplies data needed for the global section of the IGES file.

  ASK The user will be prompted for the name assigned by the receiver to the product represented by the file.

  user string The name assigned by the receiver to the product represented by the file.

  NONE If no other information is given, the string NONE will be written.

  Default: NONE

- SET W-ORGANIZATION {user-string <NONE>}

  The parameter W-ORGANIZATION supplies data needed for the global section of the IGES file.

  user-string The name of the organization or group with whom the file's author is associated.

  NONE If no other information is given, the string NONE will be written.

  Default: NONE

- SET W-AUTHOR {ASK |user-string <NONE>}

  The parameter W-AUTHOR supplies data needed for the global section of the IGES file.

  ASK The user will be prompted to enter the name of the person responsible for the creation of the data contained in the file.

  user-string The name of the person responsible for the creation of the data contained in the file.

  NONE If no other information is given, the string NONE will be written.

  Default: NONE

- SET W-EXPLODE-EXT-MAST-INST {YES | <NO>}

  This parameter enables you to handle the translation of external references (instances of the external master).

  YES Each entity will be translated and written to the IGES file.

  NO A group will be created from those entities and it will be written to the IGES file as an internal master with base point = (0,0), unit matrix and the name GR*NN, where NN is the ID of the group.

  Default: NO

Statements for Text Conversion

- MAP TEXT {iges-font} {cim-font} [{char-map-area-number}]
  MAP W-TEXT {iges-font} {cim-font} [{char-map-area-number}]

  Maps the source (IGES/Cimaton) font with the equivalent destination (IGES/**Cimatron**) font. For Write-IGES, this maps the **Cimatron** font with the equivalent IGES font. Characters from the source font are mapped to destination font characters using the character mapping file (**.cdt** file) and the char-map-area-number. Character mapping files are located in the directory **<root_cad>**\dat\fonts. The char-map-area-number is a hexadecimal number indicating the range of source font character codes to be used from the character mapping file when the given source font characters are mapped. Each char-map-area-number specifies a range of 256 characters. The **.cdt** files used by IGES Utilities are:

  wiges.cdt - for the Write-IGES option

iges-font	IGES Font Code number of the IGES Font
cim-font	**Cimatron** Font name, one of the fonts located in the directory **<root_cad>\dat\fonts**.
char-map-area-number	Area number (Hexadecimal number) in the character mapping file (**.cdt** file).

  Defaults:
  If the MAP TEXT statement for a specific font is omitted:
  Default **Cimatron** font for Read-IGES : standard
  Default IGES font for Write-IGES : 1
  Default char-map-area-number if omitted : 0

Statement for Line-Font Conversion

- MAP W-LINE-FONT {iges-line-font(1-5)} {cim90-line-font(1-8)}

  Map the **Cimatron** line font to the equivalent IGES line font.

  Defaults:
  1 - **Cimatron** Solid line font - 1 for IGES line font Solid 1
  2 - **Cimatron** Dash line font - 2 for IGES line font Dashed 2
  3 - **Cimatron** Center line font - 3 for IGES line font Centerline 4
  4 - **Cimatron** Solid line L Dash - 4 for IGES line font Dashed 2
  5 - **Cimatron** Font 5 for IGES line font Dotted 5
  6 - **Cimatron** Font 6 for IGES line font Dotted 5
  7 - **Cimatron** Font 7 for IGES line font Dotted 5
  8 - **Cimatron** Font 8 for IGES line font Dotted 5

Additional Write-IGES Utility Statements

Set the type of IGES file that will be generated:

- SELECT W-FILE-TYPE {<FORMATTED> | UNFORMATTED}

Set preferred representation of Curve On a Parametric Surface (Type 142) to be written to the IGES file.

- SET W-BOUNDARY-PREF-TYPE { GEOM |<PARAM> | EQUAL | GEOM-ONLY | PARAM-ONLY}

Approximate surface to Rational B-spline surface (type 128):

- SELECT W-SURFACE-AS-128 {ALL | TRIMMED | PRIMITIVE | <NONE>}

Approximate curve to B-spline curve (type 126):

- SELECT W-CURVE-AS-126 {<NONE> | ALL | TRIMBOUND | CURVE | SPLINE}

Set tolerance, in mm, used when a **Cimatron** entity will be approximated by an IGES entity:

- SET W-TOL {approximation-tolerance(0.001-100) <0.1>}

Approximate Surfaces to 114:

- SELECT W-SURFACES-AS-114 {PRIMITIVE | TRIMMED | ALL | <NONE>}

Preferred boundary curve representation:

- SELECT W-BOUNDARY-CURVE-TYPE {<SMOOTH> | POLYGON}

Explode SOLID model to surfaces, boundaries, or both:

- SELECT W-SOLID EXPLODE-AS {SURFACES | BOUNDARIES | <BOTH>}

Explosion of the SOLID model's planar-faces to planar-faces or planar-surfaces:

- SELECT W-SOLID-PLANAR-FACES-AS {SURFACES | <FACE>}
- SELECT W-COLOR-ENTITY {NO | <YES>}
- SELECT W-EXPLODE-EXT-MAST-INST {<NO> | YES}

  Automatically explode instances of external masters.

  The group entity in Cimatron (class 14, type 9) and it's instance (class13, type 1) will be translated as an internal master (class 14, type 1).

  In the IGES file the internal master will be written as Subfigure Definition Entity (type 308) and it's instance will be written as Singular Subfigure Instance Entity (type 408).

  Default: NO

Data to be written in the global-section of the IGES file:

- SET W-SENDER-PRODUCT-ID {user-string <NONE>}
- SET W-RECEIVER-PRODUCT-ID {ASK | user-string <NONE>}
- SET W-ORGANIZATION {user-string <NONE>}
- SET W-AUTHOR {ASK | user-string <NONE>}

Statements for Text Conversion

Map IGES fonts with equivalent **Cimatron** fonts and map the character area.

- MAP TEXT {iges-font} {cim-font} [{char-map-area-number}]
- MAP R-TEXT {iges-font} {cim-font} [{char-map-area-number}]
- MAP W-TEXT {iges-font} {cim-font} [{char-map-area-number}]

Selective Translation Statements for Read-IGES Utility

These optional statements allow partial translation of IGES entities to **Cimatron** entities. For further information, refer to the section **Selective Translation of IGES Files to Cimatron Files** in this chapter.

SELECT R-APPLICATION [NOT] {MODEL}
SELECT R-APPLICATION [NOT] {VIEW [view-number(s)]}
SELECT R-APPLICATION [NOT] {VIEW-POINTER [view-pointer(s)]}
SELECT R-APPLICATION [NOT] {DRAWING [drawing-pointer(s)]}
SELECT R-ATTRIBUTES [NOT] {TYPE IGES-type(s)}
SELECT R-ATTRIBUTES [NOT] {LEVEL level-numbers(s)}
SELECT R-ATTRIBUTES [NOT] {LINE-FONT line-font-name(s)}

```
SELECT R-ATTRIBUTES   [NOT] {COLOR color-name(s)}
SELECT R-ATTRIBUTES   [NOT] {LINE-WEIGHT line-weight-numbers(s)}
SELECT R-ATTRIBUTES   [NOT] {BLANK}
SELECT R-ATTRIBUTES   {FILTER {<AND> | OR }}
SELECT R-ENTITIES     [NOT] {entity-pointer(s)}
```

Selective Translation Statements for Write-IGES Utility

These optional statements allow selective translation of **Cimatron** entities to IGES entities. For further information, refer to the section **Selective Translation of Cimatron Files to IGES Files** in this chapter.

```
SELECT W-APPLICATION [SPLIT | NOT] {MODEL}
SELECT W-APPLICATION [SPLIT | NOT] {VIEW [view-name(s)]}
SELECT W-APPLICATION [SPLIT | NOT] {DRAWING [drawing-name(s)]}
SELECT W-ATTRIBUTES  [NOT] {LEVEL level-name(s)}
SELECT W-ATTRIBUTES  [NOT] {LINE-FONT line-font-name(s)}
SELECT W-ATTRIBUTES  [NOT] {COLOR color-number(s)}
SELECT W-ATTRIBUTES  [NOT] {PEN pen-numbers(s)}
SELECT W-ATTRIBUTES  [NOT] {BLANK}
SELECT W-ATTRIBUTES  [NOT] {SET SURFACES }
SELECT W-ATTRIBUTES  {FILTER {<AND> | OR}}
```

Write-IGES Utility Interaction Statements

Select mode of operation:

- SELECT W-MODE {<ASK> | INTER | BATCH}

Graphical Interaction Description

The Write operation may be run in either interactive or batch mode. The available options are:

ONE FILE Run the Read/Write operation interactively, one file at a time.

BATCH FILE Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

Read the part file names from a list in a batch file.

MULTI FILES Run the Read/Write operation interactively, multiple files at a time.

Select log file mode:

- SELECT W-LOG-FILE {YES | NO | <ASK>}

Graphical Interaction Description

OUT TO LOG (YES) All translation data may be saved to a log file

The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

OUT TO SHELL (NO) Translation data will be displayed on screen.

Select start-section information mode:

- SELECT W-START-SECTION {<NO> | ASK | YES} [start-section-file <iges-file-name.wis>]

Logical Operation Rules

Figure 11-1 shows the use of logical operations with filter statements.

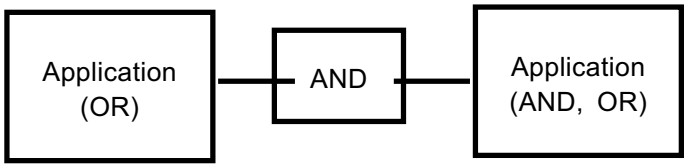


Figure 11-1: Logical Operations with Filter Statements

Figure 11-2 gives examples of the logical operation rules as applied to various filter statements. These examples will be referred to in the following explanations.

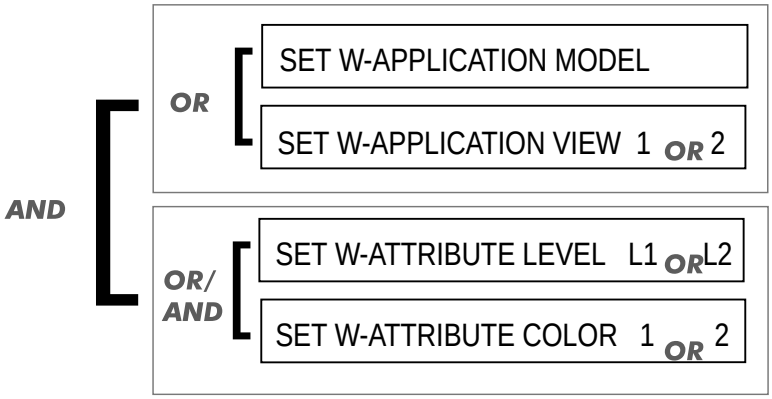


Figure 11-2: Examples of Logical Operations Applied to Filter Statements

The OR operation is performed on all argument values of the same filter argument. For example, in the statement SET R-APPLICATION VIEW 1 2, the OR operation is applied. This results in translation of entities that are visible in view 1, or visible in view 2, or visible in both views. If there is more than one statement with the same filter argument, the argument values from the different statements are treated as if they are all in one statement.

The OR operation is also applied between application filters which have different filter arguments, as shown in Figure 11-2. For example, defining the MODEL and VIEW filter statements shown in Figure 11-2 will cause translation of entities visible in the model, or visible in views 1 or 2, or visible both in the model and in views 1 and 2.

By default, the AND operation is applied between different types of attribute filters. In order for an entity to be translated, it must match at least one argument value, in each of the attribute filter statements.

To set the default to the OR operation, specify SELECT R-ATTRIBUTES FILTER OR in the parameters file. The format of the statement is:

SELECT R-ATTRIBUTES {FILTER {<AND> | OR}}

The AND operation is applied between filters of different types, i.e. between application and attribute filters.

An entity filter has no logical operations with other filters. When an entity filter is specified, the specified entity or entities are translated or not according to the entity filter exclusively.

Clarifications and Constraints

- Subordinate entities are never read by themselves, but as part of their parent entity, i.e. only independent parent entities can be filtered or unfiltered.
- Subfigure Definition (Type 308) and Network Subfigure Definition entity (Type 320) entities are always read.
- Associativity Instance entity (Type 402 Forms 1,7,9,13,14,15,16) - i.e. the forms that include geometric entities - are not filtered themselves. User filters are applied on the subordinate entities. As a result, the Associativity Instance entity is always read, but only part of it is translated, depending on what user-filters are applied.
- Filtering of Non-structure entities (Types 1xx, 2xx) is performed currently according to the parent entity regardless of the Hierarchy flag of the parent entity.
- The maximum number of argument values accepted for the COLOR filter is 9, and for the LINE-FONT filter is 6.
- During translation of the entities from the part-file, each independent parent entity is checked against the filters and the entity is filtered or not according to its entity-pointer, visibility and attribute values.

USE FILE Statement - Nested Parameter File

This statement instructs the program to include a file which contains parameter statements during the processing of the parameter (**.ipr**) file. This enables the user to nest files of parameter statements.

The statement syntax is:

USE FILE {file-pathname}

- File-pathname is full absolute file name in the file system of the included file. There are no limitations on the name, extension or location of the included file.
- The included file is processed as soon as the USE FILE statement is encountered. The rule for multiple occurrences of the same statement is applied as if all processed statements from all nested files are in one parameter file.

As an example of how this feature can be used, consider a multiple user environment. Since certain information should be the same for each user, a system parameter file that defines all the global system parameters can be written. At the same time, certain information will be specific to each user (such as the author's name). To meet both needs, the user can create his own parameter file,

which contains user-specific information, and also contains a USE FILE statement which includes the system parameter statements.

Selective Translation of Cimatron Files to IGES Files

Selective translation allows selective writing of data from the **Cimatron** database to a file in the IGES format. This is accomplished by applying user-defined filters which control whether or not a particular entity is written to the IGES file.

User Input

Figure 11-3 shows the influence of user-defined filters.

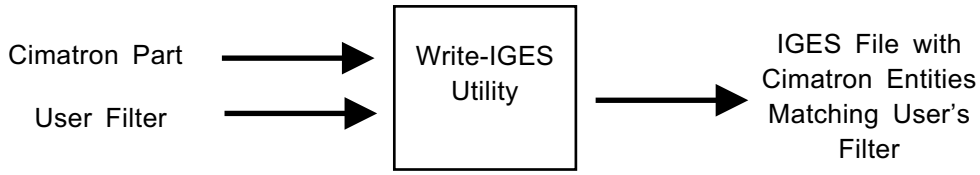


Figure 11-3: Influence of User-Defined Filters on the Write-IGES Utility

By default, the **Write-IGES** utility has no filters. All entities in the given part file are translated to the IGES file.

User-defined filter statements are placed in the parameter file (**.ipr**). Two types of filters are available:

- Application:
  model, view, drawing
- Attribute:
  level, line-font, color, pen, blank, set

A filter causes explicit and implicit inclusion or exclusion of corresponding values. Explicit inclusion/exclusion is the direct result of the filter. Implicit inclusion/exclusion results from the influence of specified argument values on argument values that were not specified in the definition.

During processing, explicit inclusion and exclusion always override implicit inclusion and exclusion. Application group filters and attribute group filters are processed separately.

Filter Syntax

Filter statements have the following syntax:

SELECT FILTER-TYPE FILTER-ARGUMENT ARGUMENT-VALUE

For example, in the filter statement SELECT W-APPLICATION DRAWING D1, the filter type is APPLICATION, the filter argument is DRAWING and the argument value is D1.

Application Filters

Application filters allow entities from specific application(s) to be included.

Following is the syntax of the application filter definition statement:

SELECT W-APPLICATION [SPLIT | NOT] MODEL

SELECT W-APPLICATION [SPLIT | NOT] {VIEW [view-name(s)]}

SELECT W-APPLICATION [SPLIT | NOT] {DRAWING [drawing-name(s)]}

Note that the filter argument MODEL takes no argument values.

Entities are included according to their visibility in the application. For example, an entity that is visible in an included view will be translated into the IGES file, even if the model in which the entity was created is not included. To give another example, including a drawing causes all views referenced within it to be written to the file.

The following rules apply:

- When no argument values are given, all available argument values for the specified application are included, and all other filter arguments are implicitly excluded.

  For example: SELECT W-APPLICATION VIEW means that all defined views in the part-file database are included, while the model and all drawings are excluded.

- When argument values are given, the specified argument values of the application are included. Other argument values for this filter argument, and other filter arguments, are implicitly excluded.

  For example: SELECT W-APPLICATION VIEW view1 view2 means that view1 and view2 are included. Other views, the model, and all drawings are excluded.

- Complement filter arguments, which begin with NOT, have different meanings according to the context.

  - When no argument values are given, all possible argument values for the specified filter argument are excluded. All other filter arguments are implicitly included.

    For example: SELECT W-APPLICATION NOT DRAWING means that all drawings are excluded, and model and all views are included.

  - When argument values are specified, all other argument values for the specified filter argument are included. The specified argument values, and all other applications are implicitly excluded.

    For example:
    SELECT W-APPLICATION NOT VIEW view1 view2 means that all views other than view1 and view2 are included, and view1, view2, the model and all drawings are excluded.

- The use of DRAWING and VIEW filters simultaneously may create a conflict. For example, when a drawing is included and one of the views within the drawing is excluded, a conflict occurs. To prevent such conflicts, the DRAWING filter and its implicit influence override the VIEW filter's explicit and implicit influence.

The effect of various filter argument and argument value combinations with the APPLICATION filter is summarized in Table 16-4.

Table 10-4. APPLICATION Filter: Explicit and Implicit Inclusion and Exclusion of Entities for Various Filter Arguments and Argument Values

Filter Arguments and Argument Value	Include	Exclude
MODEL	model	all views all drawings
VIEW	all views	model * all drawings
VIEW V1 ... Vn	views V1 ... Vn	model * views not in V1 ... Vn all drawings
DRAWING	all drawings	model ** all views **
DRAWING D1 ... Dn	drawings D1 ... Dn	model all views drawings not in D1 ... Dn
NOT MODEL	all views all drawings	model
NOT VIEW	model * all drawings	all views
NOT VIEW V1 ... Vn	views not in V1 ... Vn	model * views V1 ... Vn all drawings
NOT DRAWING	model ** all views **	all drawings
NOT DRAWING D1 ... Dn	drawings not in D1 ... Dn	model ** all views ** drawings D1 ... Dn

- Split the PFM file to many output files. This option enables you to write the model,each view, or each drawing of a PFM file to a different file.
  The output files are placed in a directory named after the PFM file
  The view/drawing file name prefix is identical to the translated view/drawing name with an addition (_d for drawing, _v for views)
  The modal file name prefix is modal. If the name of the view or drawing contains special characters such as a space ( ), dot (.),
  dollar sign ($), etc., these will be converted to an underscore (_).

Note:

- Entities are included according to their existance in the application. For example, an entity that is visible in an included view will be translated into the IGES file, even if the model in which the entity was created is not included.

* All entities that were deleted or changed in the views.

** All entities that were deleted or changed in the drawing.

Attribute Filters

Attribute filters allow entities with specified attributes to be included. For example, specifying a color will cause entities with this color to be included.

Following is the syntax of the attribute filter definition statement:

```
SELECT W-ATTRIBUTES [NOT] {LEVEL level-name(s)}
SELECT W-ATTRIBUTES [NOT] {LINE-FONT line-font-name(s)}
SELECT W-ATTRIBUTES [NOT] {COLOR color-number(s)}
SELECT W-ATTRIBUTES [NOT] {PEN pen-numbers(s)}
SELECT W-ATTRIBUTES [NOT] {BLANK}
SELECT W-ATTRIBUTES [NOT] {SET SURFACES}
```

The following rules apply:

- Argument values must always be given. The specified argument values are included, and all other argument values for the filter argument are excluded. If an attribute filter statement is given without argument values, such as SELECT W-ATTRIBUTES LEVEL, an error message will appear.
- Complement filter arguments, which begin with NOT, cause exclusion of the specified argument values and implicit inclusion of all other possible argument values.

  For example: SELECT W-ATTRIBUTES NOT LINE-FONT SOLID DOTTED means inclusion of all line-fonts other than SOLID or DOTTED.
- Line-font-name refers to the names of line-fonts displayed on the **Cimatron** line-font menu: SOLID, DASH, CENTER, L_DASH, FONT_5, FONT_6, FONT_7, FONT_8 (refer to the LINATT function).
- Color-number refers to the position of the color in the **Cimatron** color menu (refer to the FILE >> SETUP >> COLOR SETUP function). The range of color numbers is 1 to 15, where the first (top) entry in the menu has a color-number of 1. For example, if red is the third entry in the color menu, then the color-number of red is 3.
- Pen-number refers to the pen numbers displayed in the **Cimatron** pen menu (refer to the LINATT function). The range of pen numbers is 1 to 8.
- The BLANK filter is useful when blanked entities are not needed in the output IGES file. SELECT W-ATTRIBUTES NOT BLANK will cause the Write-IGES utility to ignore blanked entities.
- The SET filter is actually a filter for classes of entities in **Cimatron**. *SET* is a predefined collection of classes. Currently, the only available set is the SURFACES set which consists of all **Cimatron** surface types.

Selective Translation for Read-IGES Utility

These optional statements allow selective translation of IGES entities to **Cimatron** entities.

SELECT R-APPLICATION [NOT] {MODEL}

SELECT R-APPLICATION [NOT] {VIEW [view-number(s)]}

SELECT R-APPLICATION [NOT] {VIEW-POINTER [view-pointer(s)]}

SELECT R-APPLICATION [NOT] {DRAWING [drawing-pointer(s)]}

Attribute Filters

Attribute filters allow entities with specified attributes to be included. For example, specifying a color will cause entities with this color to be included. Following is the syntax of the attribute filter definition statement:

SELECT R-ATTRIBUTES [NOT] {TYPE IGES-type(s)}
SELECT R-ATTRIBUTES [NOT] {LEVEL level-numbers(s)}
SELECT R-ATTRIBUTES [NOT] {LINE-FONT line-font-name(s)}
SELECT R-ATTRIBUTES [NOT] {COLOR color-name(s)}
SELECT R-ATTRIBUTES [NOT] {LINE-WEIGHT line-weight-numbers(s)}
SELECT R-ATTRIBUTES [NOT] {BLANK}

- Argument values must always be given. The specified argument values are included, and all other argument values for the filter argument are implicitly excluded. If an attribute filter statement is given without argument values, such as SELECT R-ATTRIBUTES LEVEL, an error message will appear.
- Complement filter arguments, which begin with NOT, cause exclusion of the specified argument values and implicit inclusion of all other possible argument values.
  For example: SELECT R-ATTRIBUTES NOT LINE-FONT SOLID DOTTED means inclusion of all line-fonts other than SOLID or DOTTED.
- The TYPE filter's argument values refer to the entity types in the IGES specification.
  For example: SELECT R-ATTRIBUTES TYPE 100 110 means read only the Circular Arc entity (Type 100) and the Line entity (Type 110).
- The level number, taken from the fifth field of the directory section, can be a level number or a pointer to a Definition Level Property (Type 406, Form 1). Treatment of both cases is the same as for a regular level.
- Line-font-name is one of the predefined line-font names in the IGES specification: SOLID, DASHED, PHANTOM, CENTERLINE, DOTTED.
  Entities with pointers to a Line Font Definition entity (Type 304) are never filtered by the LINE-FONT filter.
- Color-name is one of the predefined color names in the IGES specification: BLACK, RED, GREEN, BLUE, YELLOW, MAGENTA, CYAN, WHITE. Entities with pointers to a Color Definition entity (Type 314) are never filtered by the COLOR filter.
- For both the LINE-FONT and COLOR filters, an UNSPECIFIED argument value is available to comply with IGES specification.

- Line-weight-number is one of the line-weight numbers as found in the IGES file to be read. These numbers depend on the specific pre-processor used to create the IGES file.
- The BLANK filter is useful when blanked entities are not needed in the **Cimatron** part-file. SELECT R-ATTRIBUTES NOT BLANK causes the Read-IGES utility to ignore blanked entities and avoid their translation.

Entity Filter

An entity filter uses a specific entity pointer value to include or exclude an entity. Following is the syntax of the entity filter definition statement:

```
SELECT R-ENTITIES [NOT] {entity-pointer(s)}:
```

The following rule applies:

Entity pointer argument values must always be given. If an attribute filter statement is given without argument values, such as SELECT R-ENTITIES, an error message will appear.

Default Parameter File and Parameter Values

The default parameter file, **default.ipr**, is located in one of the directories in the **Cimatron** file search path (the **Cimatron** file search rules are given at the beginning of this chapter). The original **Cimatron** default parameter file is located in the **<root_cad>/dat** directory. Authorization to modify this file should be restricted to the **Cimatron** system administrator.

Following is the default parameter file:

```
SELECT R-MODE ASK
SELECT R-LOG-FILE ASK
SELECT R-START-SECTION NO
SELECT R-DIMENSION-AS GROUP
SELECT R-BOUNDARY-PREF-TYPE EQUAL
SELECT R-EXPLODE GROUP NONE
SELECT R-PLANAR-SPL-AS-2D NO
SELECT R-ALL-SURFACE-TO-MODEL NO
SELECT R-FIX-TRMSRF VERIFY
SELECT R-CHECK-WEIGHTS NO
SELECT R-PEN-WIDTH-SET FALSE
SELECT R-RECON-TRS-DATA ONLY_FAILED
SELECT R-ALL-LEVELS-ON NO
SELECT R-SAFE_MODE OFF
SELECT R-DELDUP-MODE OFF
SELECT R-SPLIT-SRF NO
SELECT R-DISP-SLAVE NO
SELECT R-UNITS FILE-UNITS
SELECT W-MODE ASK
SELECT W-LOG-FILE ASK
SELECT W-START-SECTION NO
SELECT W-FILE-TYPE FORMATTED
SELECT W-BOUNDARY-PREF-TYPE PARAM
SELECT W-BOUNDARY-CURVE-TYPE SMOOTH
SELECT W-SURFACE-AS-128 NONE
```

```
SELECT W-CURVE-AS-126 NONE
SELECT W-EXPLODE-EXT-MAST-INST NO
SELECT W-SURFACE-AS-114 NONE
SELECT W-SOLID EXPLODE-AS BOTH
SELECT W-SOLID-PLANAR-FACE-AS FACE
SELECT W-COLOR-ENTITY YES
SELECT W-DIMENSION-ASSOCIATIVITY NO
SET W-TOL 0.1
SET W-SENDER-PRODUCT-ID NONE
SET W-RECEIVER-PRODUCT-ID NONE
SET W-ORGANIZATION NONE
SET W-AUTHOR NONE
SELECT W-TOL-UNITS MM
SET R-SPECIAL-TEXT -1
SET R_PREDEFINED-LEVELS-NUM 0
MAP TEXT 1001 STANDARD 1
MAP TEXT 1002 STANDARD 2
MAP TEXT 1003 STANDARD 3
MAP TEXT 3001 STANDARD 4
```

Parameter File Check

The IGES Utilities provides an option to check the format and statement syntax of the parameter file.

To check a file, use option **<3> PARAMETER FILE CHECK**, from the IGES Utilities menu.

If no errors are found in the parameter file, the following message will appear:

```
PARAMETER FILE WAS SUCCESSFULLY PARSED
```

The parameter file check does not verify statement arguments. For example, if the statement SELECT R-ATTRIBUTES COLOR 3 appears in the parameter file, and there are no entities with a color-number of 3 in the IGES file being read, you will not receive an error message. ❑

Chapter 12
IGES File Information Utility

The IGES File Information Utility provides the user with a brief description of the IGES file and a report on its validity. The utility reads the whole IGES file and generates a report file (**.irt**).

Running the IGES File Information Utility

The IGES File Information Utility may be run from the graphic or textual interaction

Graphic

How To:

1. Select Application.
2. Select IGES.
3. Select PFM Info.

Textual

How To:

1. Select option PFM FILE INFORMATION.
2. Type the name of the PFM file when prompted.
3. Messages will be displayed as the report file (**.irt**) is completed.

Notes:

- The report file created will be in the same directory and have the same name as the IGES file, but will have the extension .**irt** rather than .**igs**.
- The IGES file may be located in the current directory or in other directories. Enter the full pathname of the IGES file if it is not located in the current directory.

Sample IGES Report File

```
Cimatron  (version  11.0)  IGES  File  Information  <version  number>
==================================================================


IGES  file  Information  :
___________


#  Views                          :
        View  DP                        View  Number
        ____                         _______
        19                                 1
        23                                 2
        27                                 3
        31                                 4

#  Drawings                       :

        Drawing  DP
        ______
        399

#  Levels                         :

        Level  Number
        _______
        0
        1

#  Line-Weights                   :

        Line-Weight  Number
        __________
        0
        1
        4

#  Colors  (as  pre-defined  in  IGES  specifications          :

        Color  Name                    Color  Index
        ______                         _______
        BLACK                1
        RED                                2
        GREEN                              3
        BLUE                               4
        YELLOW                             5
        MAGENTA                            6
        CYAN                               7
        WHITE                              8

#  Line-Fonts  (as  pre-defined  in  IGES  specifications)     :

        Line-Font  Name            Line-Font  Index
        ___________             ________
        SOLID                            1
        DASHED                           2
        PHANTOM                          3
        CENTERLINE                       4
        DOTTED                           5
```

```
  IGES  File  Contains  the  Following  Data:
  ==========================================
CIRCULAR  ARC           TYPE   100          -  NUMBER  OF  ENTITIES:  9
COPIUS  DATA            TYPE   106/11       -  NUMBER  OF  ENTITIES:  2
COPIUS  DATA            TYPE   106/31       -  NUMBER  OF  ENTITIES:  1
COPIUS  DATA            TYPE   106/40       -  NUMBER  OF  ENTITIES:  23
LINE                    TYPE   110          -  NUMBER  OF  ENTITIES:  37
TRANSFORMATION  MATRIX  TYPE   124          -  NUMBER  OF  ENTITIES:  11
ANGULAR  DIMENSION      TYPE   202          -  NUMBER  OF  ENTITIES:  3
DIAMETER  DIMENSION     TYPE   206          -  NUMBER  OF  ENTITIES:  2
GENERAL  NOTE           TYPE   212          -  NUMBER  OF  ENTITIES:  27
LEADER  (ARROW)         TYPE   214/1        -  NUMBER  OF  ENTITIES:  4
LEADER  (ARROW)         TYPE   214/2        -  NUMBER  OF  ENTITIES:  33
LEADER  (ARROW)         TYPE   214/4        -  NUMBER  OF  ENTITIES:  4
LEADER  (ARROW)         TYPE   214/12       -  NUMBER  OF  ENTITIES:  5
LINEAR  DIMENSION       TYPE   216          -  NUMBER  OF  ENTITIES:  5
RADIUS  DIMENSION       TYPE   222          -  NUMBER  OF  ENTITIES:  1
GENERAL  SYMBOL         TYPE   228          -  NUMBER  OF  ENTITIES:  12
COLOR  DEFINITION       TYPE   314          -  NUMBER  OF  ENTITIES:  8
ASSOCIATIVITY  INSTANCE TYPE   402/1        -  NUMBER  OF  ENTITIES:  2
ASSOCIATIVITY  INSTANCE TYPE   402/3        -  NUMBER  OF  ENTITIES:  6
DRAWING  ENTITY         TYPE   404          -  NUMBER  OF  ENTITIES:  1
VIEW                    TYPE   410          -  NUMBER  OF  ENTITIES:  4
```

❑

Chapter 13
PFM File Information Utility

The PFM File Information Utility provides you with a brief description of the PFM file. The utility reads the PFM file and generates a report file (**.pfr**).

The report includes:

1. Names of levels within the PFM file.
   ON = The level is visible in the modal.
   OFF = The level is not visible in the modal.
2. Names of views within the PFM file.
3. Names of drawings within the PFM file.
   Names of views included in each drawing.
4. Entity statistics sections display the types and quantities of entities in the PFM file.

Running the PFM File Information Utility

The PFM File Information Utility may be run from the graphic or textual interaction.

Graphic

How To:

1. Select Application.
2. Select DXF.
3. Select PFM Info.

Textual

How To:

1. Select option PFM FILE INFORMATION.
2. Type the name of the PFM file when prompted.
3. Messages will be displayed as the report file (**.pfr**) is completed.

Notes:

- The report file created will be in the same directory and have the same name as the PFM file, but will have the extension **.pfr** rather than **.pfm.**
- The PFM file may be located in the current directory or in other directories. Enter the full pathname of the PFM file if it is not located in the current directory.

```
Cimatron  (version  9.0)  PFM  File  Information  (version  number)
==================================================================
PFM  File  Contains  the  Following  Levels:
____________________
1   - ON     2    -  OFF      3     -  OFF        4     -  OFF        5     -  OFF
6   - OFF    7    -  OFF      8     -  OFF        9     -  OFF        10    -  OFF
11  - OFF    12   -  OFF      13    -  OFF        14    -  OFF        15    -  OFF
16  - OFF    17   -  OFF      18    -  OFF        19    -  OFF        20    -  OFF
21  - OFF    22   -  OFF      23    -  OFF        24    -  OFF        25    -  OFF
26  - OFF    27   -  OFF      28    -  OFF        29    -  OFF        30    -  OFF
31  - OFF    32   -  OFF      33    -  OFF        34    -  OFF        35    -  OFF
36  - OFF    37   -  OFF      38    -  OFF        39    -  OFF        40    -  OFF
41  - OFF    42   -  OFF      43    -  OFF        44    -  OFF        45    -  OFF
46  - OFF    47   -  OFF      48    -  OFF        49    -  OFF        50    -  OFF
WRK   -  OFF

PFM  File  Contains  the  Following  Views:
____________________
        V1
        V2
        V3
        V4

PFM  File  Contains  the  Following  Drawings:
____________________
        D1

Views  included  in  drawing:
        V2
        V1
        V3
        V4

PFM  File  Contains  the  Following  Data:
____________________
LINE                  Class  1    Type  1    Number  of  Entities  :  33
ARC                   Class  1    Type  2    Number  of  Entities  :  6
LINEAR  DIMENSION     Class  5    Type  1    Number  of  Entities  :  5
RADIAL  DIMENSION     Class  5    Type  2    Number  of  Entities  :  3
ANGULAR  DIMENSION    Class  5    Type  3    Number  of  Entities  :  3
ORDINAT  DIMENSION    Class  5    Type  4    Number  of  Entities  :  9
SECTION  LINE         Class  6    Type  1    Number  of  Entities  :  2
HATCH                 Class  6    Type  2    Number  of  Entities  :  1
ID_NUM                Class  6    Type  7    Number  of  Entities  :  3
INSTANCE              Class  13   Type  1    Number  of  Entities  :  4
```

❑

Section IV

PTC Utilities

Chapter 14
Introduction to the PTC Utilities

The **Cimatron PTC Utility** inputs a file in the **PTC** (**.neu**) (Neutral Format) and converts it to a **Cimatron** part file (**.pfm**).

Only surface and curve entities may be converted to a **Cimatron** part file.

The following two utilities are included in the **Cimatron PTC Utilities**:

Read-PTC	Convert files that are in the **PTC** (Neutral Format) to files in **Cimatron** part file (**.pfm**) format, as solid or wireframe, according to the parameters file.
PARAMETER FILE CHECK	Verify a **PTC** parameter file (**.ppr**).

Running PTC

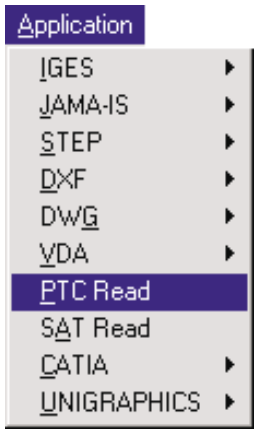
There are two methods of running the PTC Utility:

- **graphically**, using the usual Windows graphical user interface,
- **textually**, entering commands at the DOS prompt.

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select PTC Read from the Application Menu:



Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following menu is displayed:

```
Available options:

1 -> Read option
2 -> Parameter File option
```

3. Select the appropriate option.

Each of these options will be explained in a chapter of this section.

About the PTC (Neutral Files)

A neutral file is a formatted text file containing information about parts and assemblies created on the PTC system.

Overview of the PTC (Neutral Files)

A neutral file will have the following:

1. A Neutral file consists of lines of ASCII text.
2. A line in a Neutral file beginning with the character “#” is a comment.
3. All other lines in a Neutral file are of the form:

   <level><field><value>

   where:

   <level> is an integer

   <field> is a name

   <value> is a string which may be present
4. A <field> must be one of the following:
   a. The name of a simple data type (integer,string,real,etc.)
   b. The name of an array.
   c. The name of a structure.
5. All the fields on a particular level belong to the same parent.
6. When <value> is present at a certain <level> and begins with the character “[”, or consists of a series of digits separated by commas, it indicates that <field> is an array of dimension [n]([m]...). The array element values are contained on the lines <level+1> up to the next line starting with <level> again.

   For one-dimensional arrays there is only one line at the higher level, whose <value> is a string of element values separated by commas.

   The notation “n*m” indicates the following “n” elements each have the value “m”.
7. When the <value> is present at a certain <level> (and is not the string “->”), then this is the value to be assigned to <field>.
8. When <value> is absent at a certain <level> this indicates that <field> is a structure. Immediate elements of the aggregate are contained on lines <level+1> up to the next line starting with <level> again.
9. A special case is the line whose <field> is the string “ugc_xar_len” indicating that the next line begins the description of an array of length <value>.
10. When <value> is the string “->” this indicates that <field> is a pointer to an aggregate whose elements follow.
11. The special <value> string “NULL” indicates that the <field> is a pointer whose value is null. ❑

Chapter 15
The Read-PTC Utility

The **Read-PTC** Utility converts files in the PTC format to files in the **Cimatron** part file (**.pfm**) format.

Read-PTC uses a parameter file (**.ppr**). This file contains parameters that control the operation mode and determine the user interaction (refer to the section **Read-PTC Parameters**).

For more information about the parameter file and the format of its statements, refer to **The PTC Utilities Parameter File**.

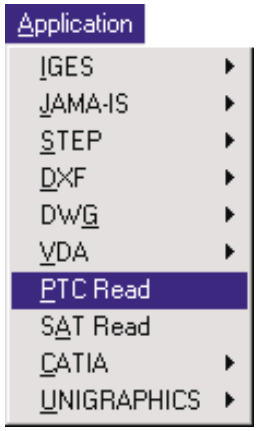
Translation data can either be displayed on the screen or optionally saved to a log file. The log file contains information such as user parameter values, a listing of each entity translated and any errors or problems encountered during its translation.

Running Read-PTC

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select PTC Read from the Application Menu:



Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following PTC options are available:

```
1  ->  Read  option
2  ->  Parameters  File  option
```

3. Select the **Read** option.

Running Modes

Read-PTC may be run interactively or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or in **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

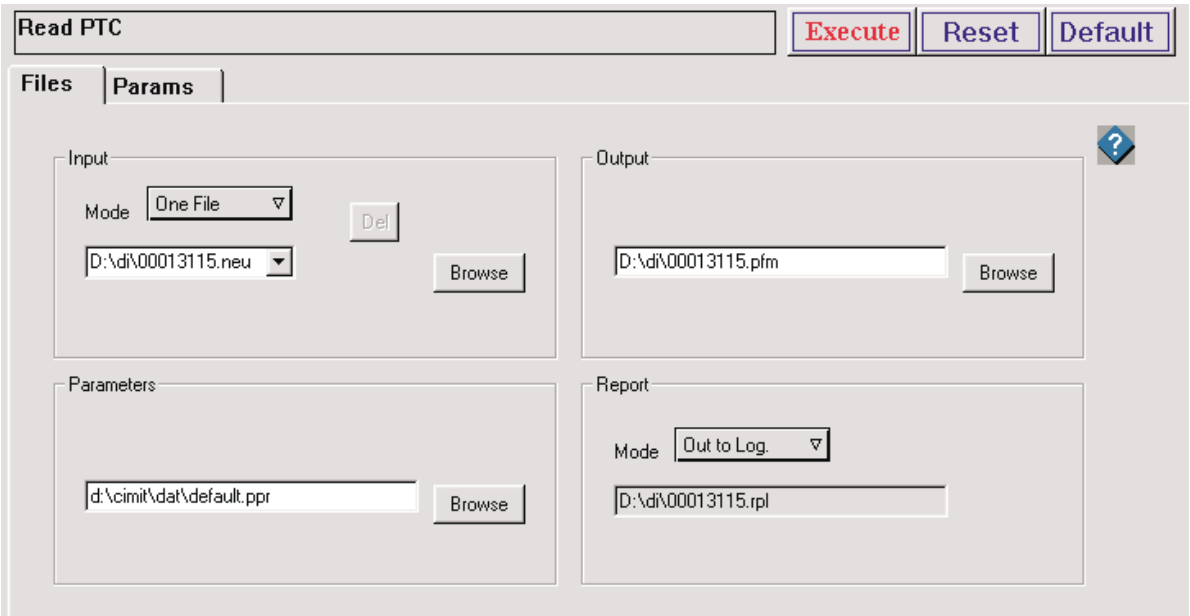
1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file containing one file name per line.

Notes:

- Multi mode is only available when you use the graphic interface for this utility.
- Each part file created will be in the same directory and have the same name as the **PTC file**, but will have the extension **.pfm** instead of **.neu**.
- The **PTC** file may be located in the current or other directories. Enter the full path name of the **PTC** file if it is not located in the current directory.
- File extensions may be typed, unless otherwise specified. If the file name is not found as it is typed, the program will search for the same file name with the extension **.neu**.
- In **Batch Mode Without Backup**, if a file already exists with the same name, including the **PTC** extension, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and **.ptc** files already exist with the name specified in the list, the old **.pfm** file will be copied to a backup file with the extension **.bac**. Then a new **.ptc** file will be created from the PTC file.
- Each **.pfm** created has two levels:
  1. CURVES - The wireframe representation.
  2. SURFACE - The surface modal representation

Graphic Interaction with Read-PTC



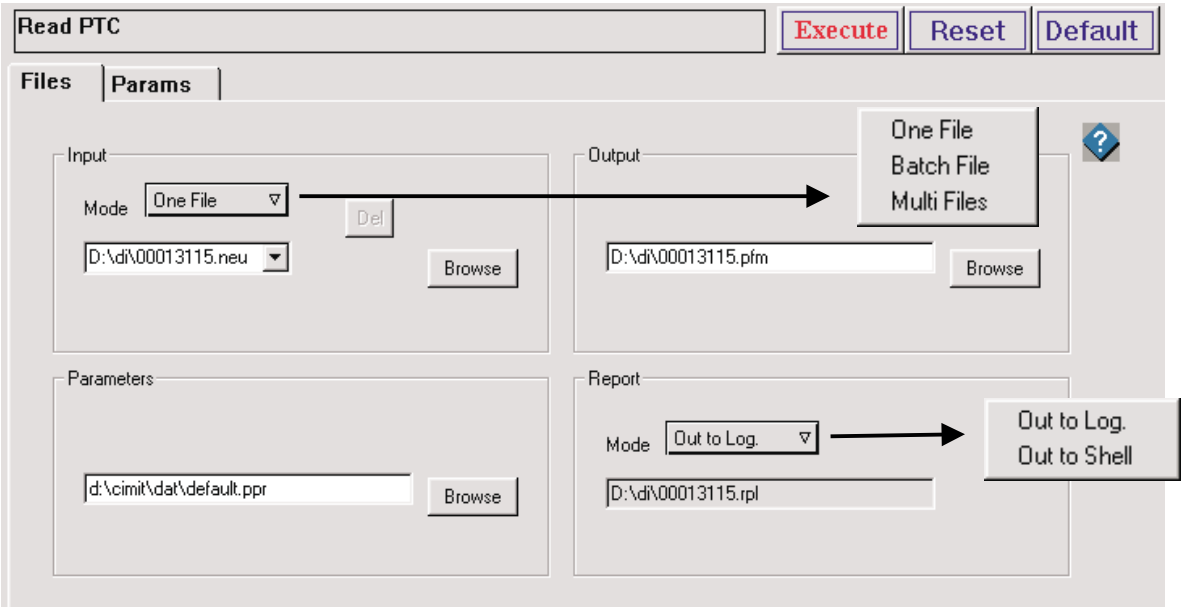
How To:

1. Select the appropriate tab: **Files** or **Params**.
2. Set the parameters.
3. Select a file in the Input area using the Browse button.
4. Press the Execute button.

The interaction for each of the tabs is described on the following pages.

Files

This dialog defines the Input, Output, Report and Parameter files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input **.neu** file (the file to be converted to a Cimatron **.pfm**).

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.pfm** file is identical to the prefix of the input **.neu** file.

   Choose a different **.pfm** file, or using the Browse button, select an existing **.pfm** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the IGES parameter file is **default.ppr.** You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read PTC translation data **rpl** file is identical to the prefix of the input **.neu** file.

   Choose a different **.rpl** file, or using the Browse button, select an existing **.rpl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

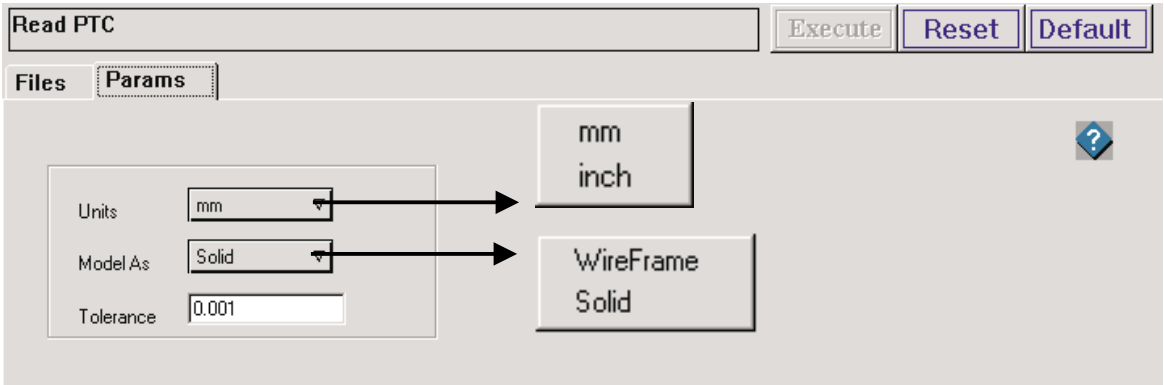
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **Units:** Set the PTC file units of measurement with those of the Cimatron file. Only mm and inch are available.
2. **Model As:** Create entities in either Wireframe (explode the model entities to wireframe) or Solid (read the model entities as solid).
3. **Tolerance:** Set the solid tolerance. The tolerance range is 0.0001-1.0 (Default=0.001). If the text in the window is deleted, the last entry is selected.

Textual Interaction with Read-PTC

The following section describes the interaction upon invoking **Read-PTC**. (The interaction that is shown will occur when all the interaction parameters in the parameter file (**.ppr**) are set to ASK.) All prompts that appear on the screen are written here in italics.

PTC (version 18.0) - Cimatron (version 11.0) Converter <version number>
==========

Enter PTC parameters file name (<CR> = default):	Type name of a parameter file; do not type the **.ppr** extension. To use the default parameter file (**default.ppr**), press <CR>.
Available modes: *1 -> Interactive mode* *2 -> Batch mode with backup* *3 -> Batch mode without backup*	The mode menu will appear.
Enter mode (0 = exit)	Type the number of the appropriate operation mode. To exit the program at this point, type 0 (zero).

If Interactive mode is selected:

Enter PTC file name (<CR> = mode menu)	Type the name of the PTC file to be converted; do not type the **.neu** extension. To exit the program at this point, press <CR>.
Enter part file name (<CR> = <file_name>) :	Type a name for the part file. To use the name displayed in parentheses, press <CR>. If the specified file already exists, the following message appears:
—- File <file_name>PTC already exists Replace existing file ? (Y/N)	To write over the existing file, type **y**. To type a different name for the file, type **n**.
Save PTC log file (Y/N) ? (<CR> = Y) :	You will be asked whether you want to save the **Read-PTC** log file. If you enter **y**, all translation data will be saved to a log file. The name of the log file will be that of the PTC file with the extension **.rpl**. If you enter **n**, the translation data will be displayed on the screen.
Setting up part file <file_name> ...	The new part file will be saved with the **.ptc** extension.

Translating entities from PTC to Cimatron:

End of entities translation
Number of edges = Number of faces =
End of entity translation
Saving part file <file name>
Saving file <file name> .rpl

❑

Chapter 16
The PTC Utilities Parameter File

The PTC Utilities parameter file allows the PTC utilities user to:

- customize the interaction with the utilities.
- insert data required by the utilities.

This chapter contains a general description of the creation, format and use of the parameter file. Further information on the **Read-PTC** parameters can be found in Chapter 15, **The Read-PTC Utility**.

A description of the **Parameter File Check** Utility is given at the end of this chapter.

Parameter File Name

The format of the parameter file name is **filename.ppr**. There are no limitations on the name, however the extension must be **.ppr**. For example, the name of the default parameter file is **default.ppr**.

Parameter File Creation

The **Cimatron PTC** parameter file is created by the user. Since the file is in ASCII (American Standard Code for Information Interchange) format, it can be easily created using any text editor. The **Cimatron** original default parameter file (**default.ppr**), located in the **<root_cad>\dat** directory, can be used as a template and guide for the user.

Parameter File Format and Rules

The parameter file is an ASCII file. The file contains statements which consist of predefined keywords and their parameter values. Following are some general rules of the parameter file format:

- The order of the lines is not important.
- Letter case is not important except when specifying **Cimatron** files, in which case the **Cimatron** file naming rules must be used.
- Each line starts with a keyword.
- If the same keyword appears more than once, and the statements can be accumulated, they will; otherwise the last statement encountered will be used.
- A line that starts with ## is a comment line.
- Empty lines are ignored.
- Maximum of 50 words are allowed in one statement.
- To continue the arguments of a statement onto an additional line, finish the current line with \<CR> (backslash followed by carriage return) and continue on to the next line.
- The maximum number of characters allowed in one statement is 256.

Following are the conventions used to describe the parameter file statements:

- | - separates different options of the same argument.
- {A} - argument is required.
- [A] - argument is optional.
- <A> - argument is the default value if no value is given.
- (n-m) - range of possible values.
- Upper case words are fixed predefined keywords or arguments values. They should be typed in the parameter file exactly as they appear here.
- Lower case words are used for variables for which the user would normally substitute a value.
- The keyword ASK means that the user will be prompted to supply information when running the program.

Read-PTC Utility Interaction Statements

Select mode of operation:

- SELECT R-MODE{<ASK> I INTER I BATCH} [BACKUPI<NOBACKUP>]

Select log file mode:

- SELECT R-LOG-FILE {YES I NO I <ASK>}

Set the PTC file units with the equivalent **Cimatron** units:

- SET R-UNITS { <MM> I INCH }

Set the PTC model to wireframe or solid:

- SELECT R-MODEL-AS {WIREFRAME | <SOLID>}

Set the solid tolerance:

- SET R-SOLID-TOLERANCE {0.0001 | 1.0}

Read-PTC Parameters

The parameter file (**.ppr**) contains parameters that control the operation mode, determine the interaction when running **Read-PTC** and provide information needed by the utility. These parameters will be described here.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP |<NOBACKUP>]

  The parameter R-MODE selects the mode of operation.

  Read-PTC may be run interactively or in batch mode. Optional backup is available with batch mode.

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. User is prompted for name of each PTC file to be converted.
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the PTC files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.

BACKUP Option used in batch mode. If a **PTC** file already exists with the name specified in the batch file, it will be copied to a backup file with the extension **.bac.**

NOBACKUP This is a system default option used in batch mode. If a part file (**PTC**) with the name specified in the list already exists the new part file will replace it.

Defaults: ASK, NOBACKUP

- SELECT R-LOG-FILE {YES | NO | <ASK>}

  The parameter R-LOG-FILE selects log file mode.

  Translation process data can either be saved to a log file or displayed on the screen.

  YES All translation data will be saved to a log file. The name of the log file will be that of the PTC file with the extension **.rpl**. The screen will display only information about the PTC type and form contained in the PTC file.

  NO Translation data will be displayed on the screen.

  ASK User is asked whether he wants to save translation data to log file.

  Default: ASK

- SELECT R-MODEL-AS {WIREFRAME | <SOLID>}

  Create entities in WIREFRAME or SOLID.

  SOLID Read the entities of the model as SOLID.

  WIREFRAME Explode the entities of the model to WIREFRAME.

  Default: SOLID.

- SELECT R-UNITS {MM | INCH}

  Sets PTC file units with equivalent **Cimatron** units.

  Default: MM

- SET R-SOLID-TOLERANCE {0.0001 | 1.0}

  Set the solid tolerance.

  Default: 0.0001

Default Parameter File and Parameter Values

The default parameter file, **default.ppr**, is located in one of the directories in the **Cimatron** file search path (the **Cimatron** file search rules are given at the beginning of this chapter). The original **Cimatron** default parameter is located in the **<root_cad>\dat** directory. Authorization to modify this file should be restricted to the **Cimatron** system administrator.

Following is the default parameter file:

```
SELECT R-MODE ASK
SELECT R-LOG-FILE ASK
SELECT R-MODEL-AS SOLID
SET R-UNITS MM
SET R-SOLID-TOLERANCE 0.001
```

Customized Default Parameter File

Each user can customize his own default parameter by creating a parameter file named **<filename>.ppr** in the directory **<root_cad>\dat**.

Parameter File Check

The PTC Utilities provides an option to check the format and syntax of the parameter file.

To check a file, use option **<2> PARAMETER FILE CHECK**, from the PTC Utilities menu. If no errors are found in the parameter file, the following message will appear:

❑ PARAMETER FILE WAS SUCCESSFULLY PARSED

The parameter file check does not verify statement arguments. ❑

Section V

VDA Utilities

Chapter 17
Introduction to the VDA Utilities

Geometric models containing points, point sets, curves and surfaces, may be transmitted in VDA format between different systems using the utilities described in this chapter.

The following three functions are included in this utility:

Read-VDA	Convert files which are in VDA format to **Cimatron** part file (**.pfm**) format.
Write-VDA	Convert Cimatron geometric part files (**.pfm**) to files in VDA format.
PARAMETER FILE CHECK	Verify a VDA parameter file (**.vpr**)

Running VDA

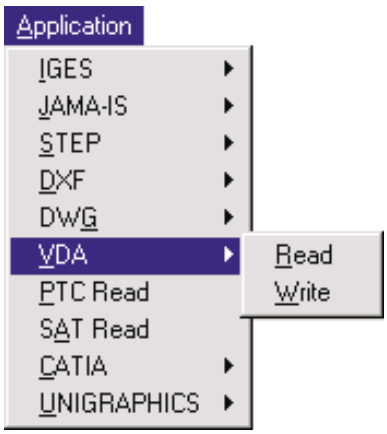
There are two methods of running the VDA Utility:

- **graphically**, using the usual Windows graphical user interface,
- **textually**, entering commands at the DOS prompt.

Graphic

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. Select VDA from the Application Menu. The following options are displayed:



3. Select the appropriate option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following VDA options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
```

3. Select the appropriate option.

Each of these utilities will be explained in a chapter of this section.

RDVDA

Convert files which are in VDA format to **Cimatron** part file format and include them in the model in the current part file. This is an internal routine and as such is only accessible via the USER function.

To convert **Cimatron** part files to VDA format, use the external Write-VDA program. When converting external files to **Cimatron** part file format, use either the external Read-VDA program or the internal RDVDA routine via the USER function (see the Utilities Manual).

External programs run in the background. The internal RDVDA program runs in the foreground and adds the converted geometric entities to the current part file.

About the VDA-Interface

The CAD/CAM Committee of the VDA (Verband der Automobilindustrie) was founded in 1982.

The VDA Surface Interface (VDAFS) for the transfer of product definition data has proven itself in the area of free-form surfaces.

The lines of the file header contain the following data:

> VDA version name of sender company, contact person,
> telephone no., address, file name, project name, etc.
>
> name=BEGINSET
>
> name=ENDSET
>
> The names of a “BEGINSET” and its “ENDSET” must be identical and all the elements defined in this set will be put by **Cimatron** in the same level.
>
> name=END

The last record in a file must be the trailer **END** . The trailer name correspond to the header name.

The total transferred data volume between header and trailer identification is designated as a file. The file consists of records, each with 80 characters in ASCII code.

The records contain positive sequence numbers right-justified in columns 73 to 80. These numbers are in ascending order.

The logical parts of the file are header, geometrical and structural elements, data transformations as well as comments and trailer.

```
CIMVDA=HEADER / 20                                                  00001000
*************************************************************         00001010
VDAFS  VERSION  2.0                                                 00001020
———————-  DATA  ABOUT  SENDER———————————-                           00001030
SENDING  COMPANY:                                                   00001040
SENDER'S  NAME:                                                     00001050
TELEPHONE  NO.:                                                     00001060
ADDRESS:                                                            00001070
SENDING SYSTEM: CIMATRON VERSION 9.0VDA                             00001080
SENDING  DATE:                                                      00001090
FILE  NAME:  VDA1                                                   00001100
———————-DATA  ABOUT  THE  PART  ———————————-                        00001110
PROJECT  NAME                                                       00001120
OBJECT  CODE                                                        00001130
VARIANT                                                             00001140
CONFIDENTIALITY:                                                    00001150
DATE  EFFECTIVE:                                                    00001160
———————-DATA  ABOUT  RECEIVER  ———————————                          00001180
RECEIV.  NAME / DEPT                                                00001190
*************************************************************         00001200
$$                                                                  00001210
SET01  = BEGINSET                                                   00001220
$$                                                                  00001230
PNT1 = POINT / 0.,0.,0.                                             00001240
$$                                                                  00001250
SET01  = ENDSET                                                     00001260
$$                                                                  00001270
SET02  = BEGINSET                                                   00001280
$$                                                                  00001290
CRV1= CURVE / 1,0.,1.,                                              00001300
2,                                                                  00001310
0.,0.,0.,0.,0.,1.                                                   00001320
$$                                                                  00001330
SET02 = ENDSET                                                      00001340
$$                                                                  00001350
SET03  = BEGINSET                                                   00001360
$$                                                                  00001370
SRF2 = SURF / 1,1,0.,1.,0.,1.,                                      00001380
7,2,                                                                00001390
500.,785.39816,0.,-322.18239,-4.28473,48.12583,-7.05686,0.,0.,0.,0.,0.  00001400
0.,0.,0.,0.,0.,0.,0.,0.,0.,-250.,0.,0.,0.,0.,0.,0.,0.,0.,-616.85028,    00001420
$$                                                                  00001430
SSET03 = ENDSET                                                     00001440
$$                                                                  00001450
CIMVDA = END                                                        00001460 ❑
```

Chapter 18
The Read-VDA Utility

The **Read-VDA** Utility converts files in the VDA format to files in the **Cimatron** part file (**.pfm**) format.

Read-VDA uses a parameter file (**.vpr**). This file contains parameters that control the operation mode and determine the user interaction (refer to the section **Read-VDA Parameters**).

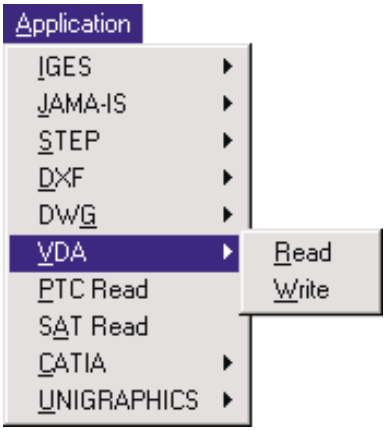
For more information about the parameter file and the format of its statements, see Chapter 20, **The VDA Utilities Parameter File**.

Running Read-VDA

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select VDA from the Application Menu. The following options are displayed:



3. Select the **VDA, Read** option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following VDA options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
```

3. Select the **Read** option.

Running Modes

Read-VDA may be run interactively or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or in **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

1. Read file names from a batch file.
2. Convert each file listed in the batch file.

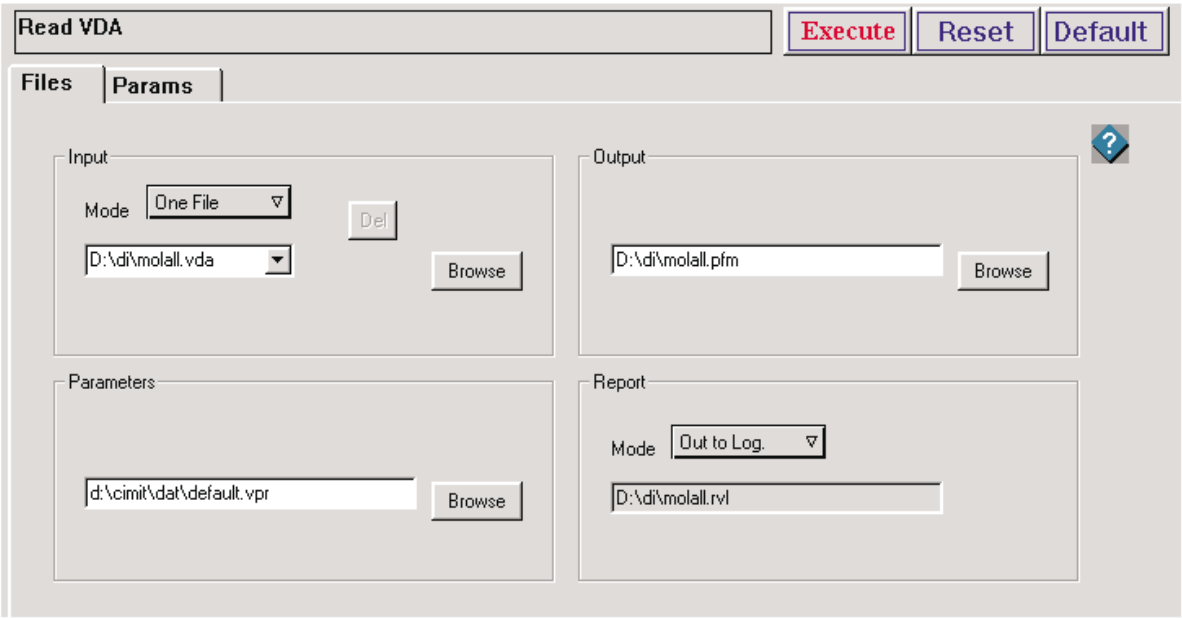
In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file containing one file name per line.

Notes:

- Multi mode is only available when you use the graphic interface for this utility.
- Each part file created will be in the same directory and have the same name as the VDA file, but will have the extension **.pfm** instead of **.vda**.
- The VDA file may be located in the current or other directories. Enter the full path name of the VDA file if it is not located in the current directory.
- File extensions may be typed, unless otherwise specified. If the file name is not found as it is typed, the program will search for the same file name with the extension **.vda**.
- In **Batch Mode Without Backup**, if a file already exists with the same name, including the **.pfm** extension, the file created with the program will replace it.

- If **Batch Mode With Backup** was selected, and **.pfm** files already exist with the name specified in the list, the old **.pfm** file will be copied to a backup file with the extension **.bac**. Then a new **.pfm** file will be created from the VDA file.
- A **Cimatron** level will be created for each VDA SET or GROUP. The **Cimatron** level will have the same name as the VDA set or group, and will have the same entities assigned to it.
- The number of surface display curves is reduced to one in order to reduce the display time.
- PSET entities are available as a string.

Graphic Interaction with Read-VDA



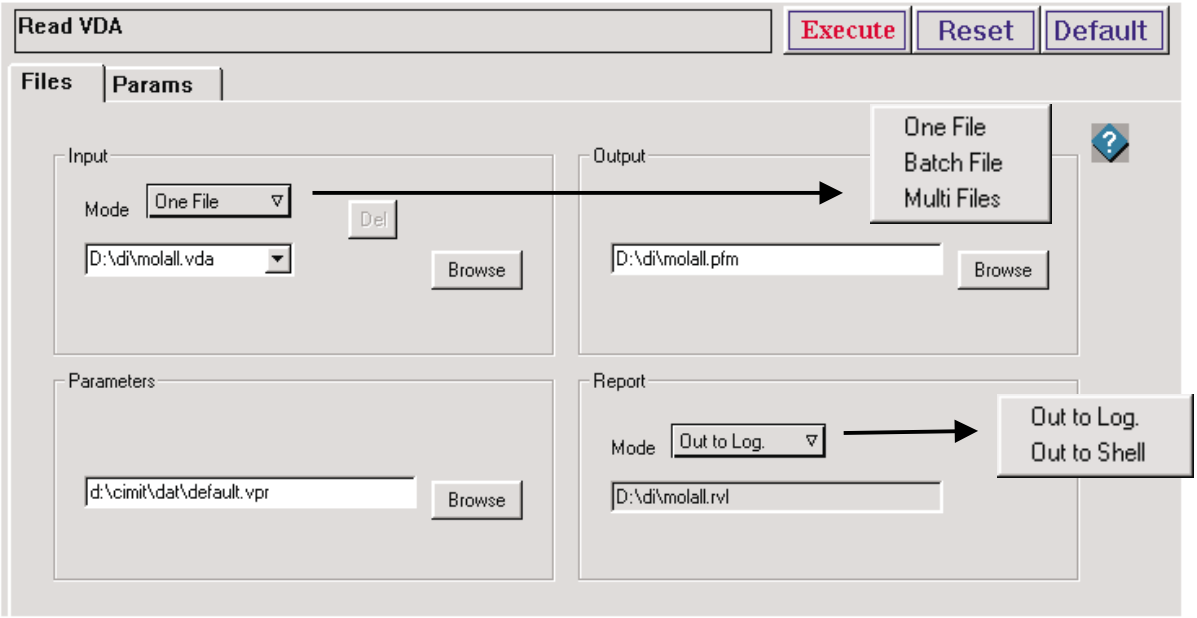
How To:

1. Select the appropriate tab: **Files** or **Params**.
2. Set the parameters.
3. Select a file in the Input area using the Browse button.
4. Press the Execute button.

The interaction for each of the tabs is described on the following pages.

Files

This dialog defines the Input, Output, Report and Parameter files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input **.vda** file (the file to be converted to a Cimatron **.pfm**).

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.pfm** file is identical to the prefix of the input **.vda** file.

   Choose a different **.pfm** file, or using the Browse button, select an existing **.pfm** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the VDA parameter file is **default.vpr.** You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read VDA translation data **rvl** file is identical to the prefix of the input **.vda** file.

   Choose a different **.rvl** file, or using the Browse button, select an existing **.rvl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

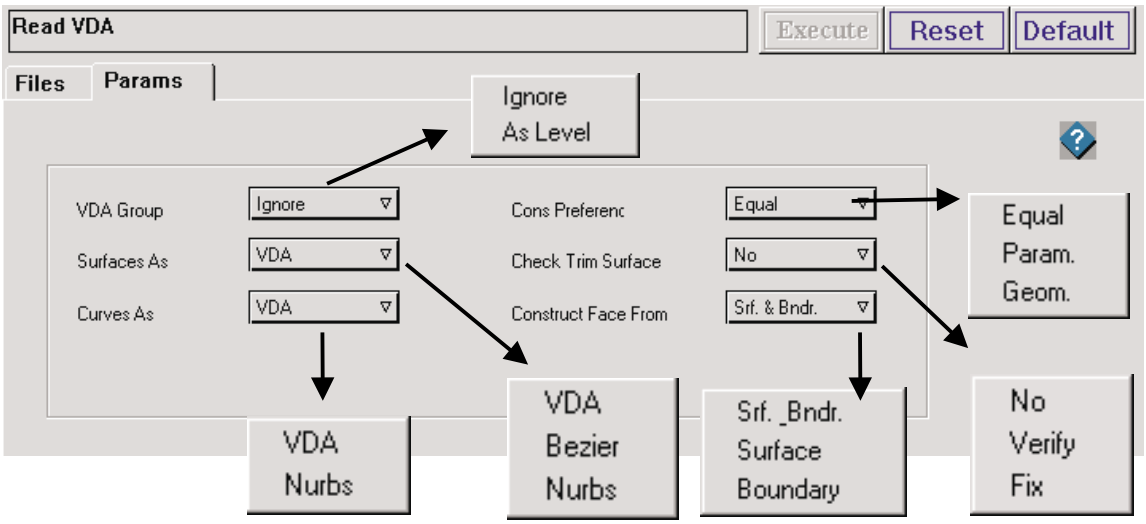
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **VDA Group:** Read VDA groups as levels or ignore them.
2. **Surfaces As:** Read VDA surfaces as VDA, Bezier or Nurbs surfaces.
3. **Curves As:** Read VDA curves as VDA or Nurbs curves.
4. **Cons Preference:** Define the preferred Curve On Surface (CONS) representation. VDA faces may be read as parametric, geometric or equal.
5. **Check Trim Surfaces:** Mode of fixing trimmed surfaces (No, Verify or Fix).
6. **Construct Face From:** When translating a surface from VDA to a Cimatron PFM, choose which entities to translate (only the Surface, only the Boundary or both the Surface and the Boundary).

Textual Interaction with Read-VDA

The following section describes the interaction upon invoking **Read-VDA**. (The interaction that is shown will occur when all the interaction parameters in the parameter file (**.vpr**) are set to ASK.) All prompts that appear on the screen are written here in italics.

VDA (version 2.0) - Cimatron (version 11.0) Convertor <version number>

==========

Enter VDA parameters file name (<CR> = default):

Type name of a parameter file; do not type the **.vpr** extension. To use the default parameter file (**default.vpr**), press <CR>.

Available modes:

The mode menu will appear.

1 -> Interactive mode
2 -> Batch mode with backup
3 -> Batch mode without backup

Enter mode (0 = exit)

Type the number of the appropriate operation mode. To exit the program at this point, type 0 (zero).

If Interactive mode is selected:

Enter VDA file name ((<CR> = Y/N)? <N> means exit) :

Type the name of the VDA file to be converted; do not type the **.vda** extension. To exit the program at this point, press <CR>.

Enter part file name (<CR> = <file_name>) :

Type a name for the part file. To use the name displayed in parentheses, press <CR>.
If the specified file already exists, the following message appears:

—- File <file_name>.pfm already exists
Replace existing file ? (y/n)

To write over the existing file, type **y**. To type a different name for the file, type **n**.

Read Group entity as level (Y/N)? <N>

The default is NO, Group is ignored and set entity is read as a level. If the answer is YES, the group entity is read as a level and the set entities are ignored.

Setting up part file <file_name> ...

The new part file will be saved with the **.pfm** extension.

Translating entities from VDA to **Cimatron**:

NAME	POINT	PSET	MDI	ARC	CURVE	SURF	CONS	FACE
<SET LEVEL 1>	0	0	0	180	426	20	25	1
<SET LEVEL 2>	0	0	0	0	0	0	0	0
TOTAL	0	0	0	180	426	20	25	1

END OF READING THE VDA FILE.

End of entity translation.

If Rend Group entity as level (Y / N)? (<CR>=N) Y

Setting up part file name...

Translating entities from VDA to **Cimatron**:

NAME	POINT	PSET	MDI	ARC	CURVE	SURF	CONS	FACE
TOTAL	0	0	0	180	426	20	25	1

List of GROUP Entities Names:

LAYER<n>
LAYER<n>
LAYER<n>
LAYER<n>
LAYER<n>

END OF READING THE VDA FILE

GROUP LAYER <n> IS CREATED AS LEVEL LAYER<n>

GROUP LAYER <n> IS CREATED AS LEVEL LAYER<n>

GROUP LAYER <n> IS CREATED AS LEVEL LAYER<n>

GROUP LAYER <n> IS CREATED AS LEVEL LAYER<n>

GROUP LAYER <n> IS CREATED AS LEVEL LAYER<n>

GROUP LAYER <n> IS CREATED AS LEVEL LAYER<n>

End of entities translation

Saving part <file_name>.

Enter VDA file name (<CR> = mode menu):

Entities that are not referenced by any group are put into LEVEL 0.

❑

Chapter 19
The Write-VDA Utility

The Write-VDA utility converts **Cimatron** part files (**.pfm**) to files in the VDA format.

Write-VDA uses a parameter file (**.vpr**).

For more information about the parameter file and the format of its statements, refer to Chapter 20, The VDA Utilities Parameter File.

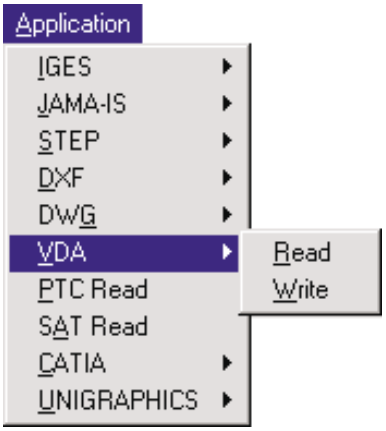
Translation data can either be displayed on the screen or optionally saved to a log file. The log file contains information such as user parameter values, global-section data written to the VDA file, a listing of each entity translated and any errors or problems encountered during translation, and statistical information on the entities contained in the **Cimatron** file.

Running Write-VDA

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select VDA from the Application Menu. The following options are displayed:



3. Select the **VDA, Write** option:

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following VDA options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
```

3. Select the **Write** option.

Running Modes

Write-VDA may be run interactively or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or: in **MULTI** mode:

1. Select multiple files.

Batch

In **BATCH** mode:

How To:

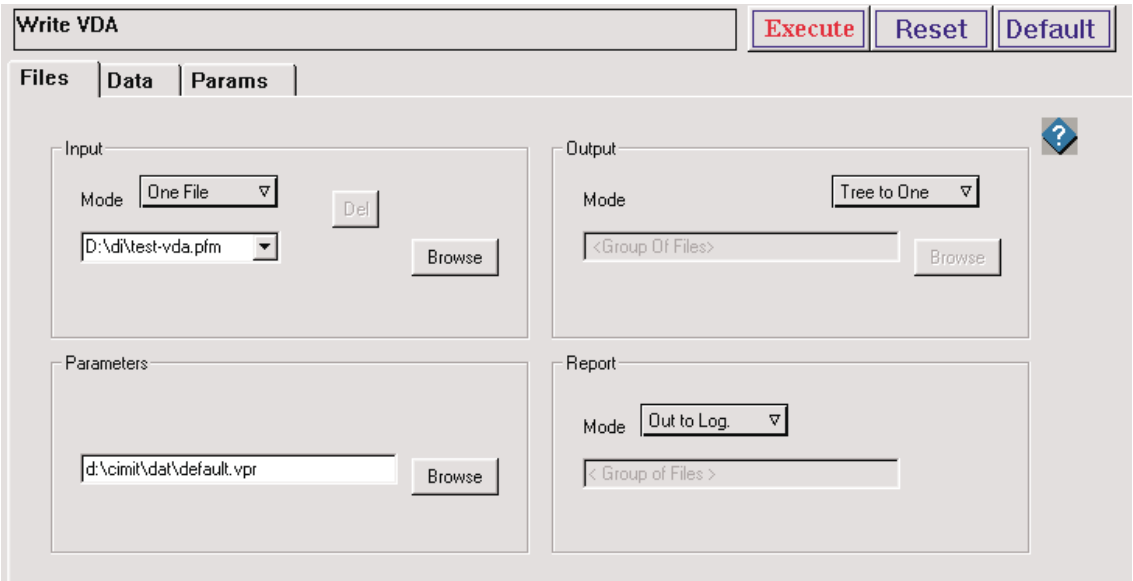
1. Read part file names from a list in a batch file.
2.. Convert each file in the list.

In **BATCH** mode a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file containing one file name per line.

Notes:

- Multi mode is only available when you use the graphic interface for this utility.
- Each VDA file created in BATCH mode, or in INTERACTIVE mode if no other name is entered, will be in the same directory and have the same name as the part file, but will have the extension **.vda** instead of **.pfm**.
- Both the part file and the generated VDA file may be located in the current or other directories. Enter the full path name of the file if it is not located in the current directory.
- Only entities which were created using the Modeling application will be converted. (Instances of masters and sub-assemblies will not be converted)
- A VDA set will be created for each **Cimatron** level. The VDA set will have the same name as the **Cimatron** level and will have the same entities assigned to it.
- In BATCH mode, if a file already exists with the same name, including the **.vda** extension, the file created with this program will replace it.
- Color- white.

Graphic Interaction with Write-VDA



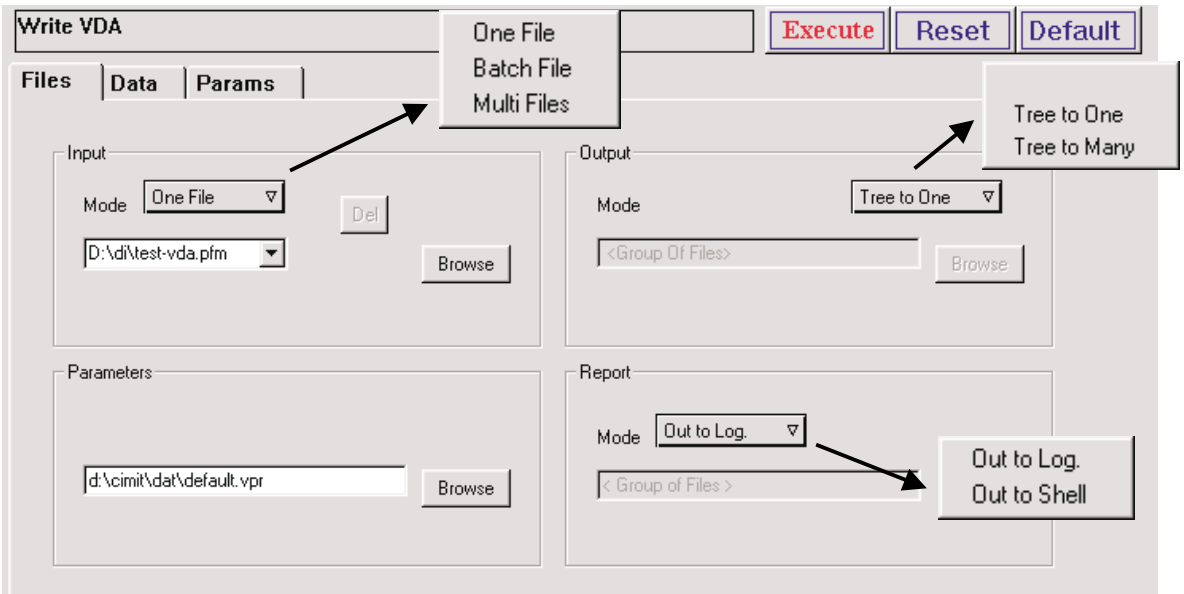
How To:

1. Select the appropriate tab: **Files, Data** or **Params**.
2. Set the parameters.
3. Select a file in the Input area using the Browse button.
4. Press the Execute button.

The interaction for each of the tabs is described on the following pages.

Files

This dialog defines the Input, Output, Report and Parameter files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Mode: One File, Batch File or Multi Files.
2. Select the input **.pfm** file (the file to be converted to a VDA file).

The available options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.vda** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.vda** file, or using the Browse button, select an existing **.vda** file. In the latter case, you will be prompted whether or not to overwrite the existing file.
2. Select an Assembly Filter: TREE TO ONE or TREE TO MANY.

The following Assembly Filters are available:

- TREE TO ONE — Create *one* VDA file for the assembly and drafting models.

  VDA drafting file name:

  assem_draft.vda.
- TREE TO MANY — Create a directory containing *many* VDA files for the assembly model; one for each part file in the assembly and one VDA file for the drafting model.

  VDA model file names:

  part1.vda, part2.vda, etc.

  VDA drafting file name:

  assem_drf.vda

Parameters

How To:

1. By default, the prefix of the VDA parameter **.vpr** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.vpr** file, or using the Browse button, select an existing **.vpr** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

The **parameter file** enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Write VDA translation data **.wvl** file is identical to the prefix of the input **.pfm** file.

   Choose a different **.wvl** file, or using the Browse button, select an existing **.wvl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

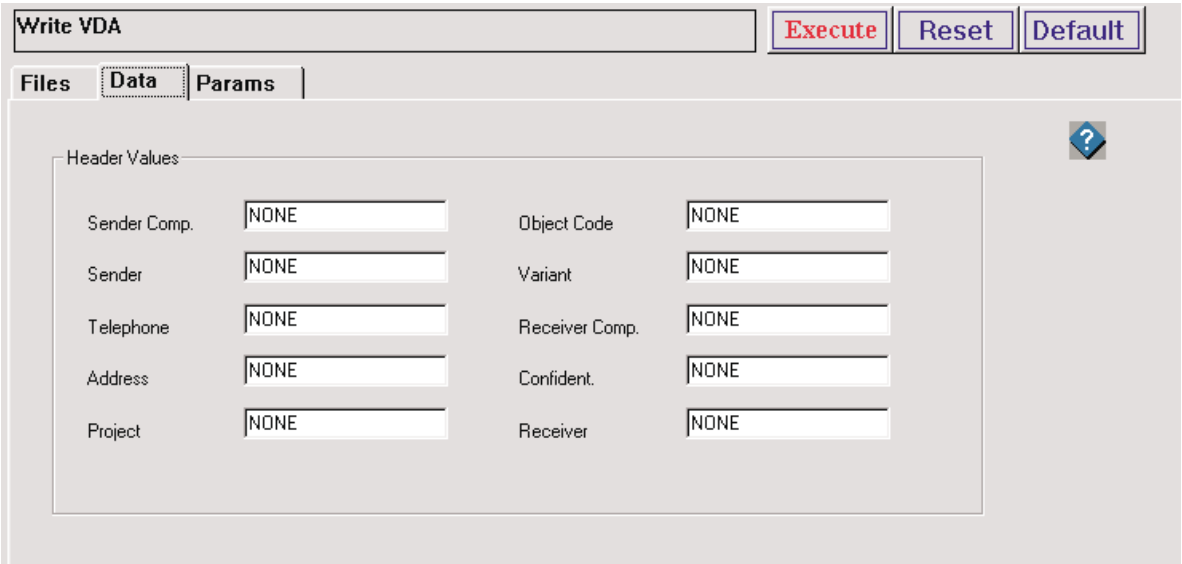
The available options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Data

This dialog defines the header values used to produce the VDA file.

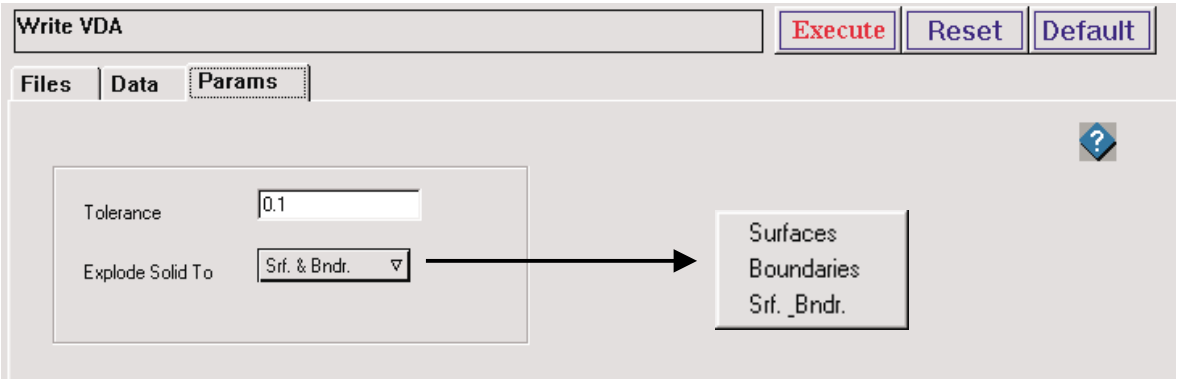


How To:

1. **Sender Comp.:** Enter the name of the sending company or leave as NONE.
2. **Sender:** Enter the name of the sender or leave as NONE.
3. **Telephone:** Enter the telephone number of the Sender or leave as NONE.
4. **Address:** Enter the address of the Sender.
5. **Project:** Enter the Part Project name.
6. **Object Code:** Enter the Part Project object code.
7. **Variant:** Enter the Part Variant.
8. **Confident.:** Enter the Part Confidentiality.
9. **Receiver Comp.:** Enter the name of the Receiving company.
10. **Receiver:** The name of the Receiver.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **Tolerance**: Set the tolerance in MM when approximating a Cimatron entity by an entity in another standard. The tolerance range is 0.001 - 100 (Default = 0.1). If the text in the window is deleted, the last entry is selected.
2. **Explode Solid To**: Explode a Solid model to Surfaces, Boundaries or Surfaces & Boundaries.
   (Default = Surfaces & Boundaries)

Textual Interaction with Write-VDA

The following section describes the interaction upon invoking **Write-VDA**. (The interaction that is shown will occur when all interaction parameters in the parameter file (**.vpr**) are set to ASK.) All screen displays are shown here in italics.

Cimatron (version 11.0) VDA (version 2.0) - Convertor <version number>
==========

Enter VDA parameters file name (<CR> = default):

Type the name of the parameter file without the **.vpr** extension. To use the default parameter file, press <CR>.

Available modes:

The mode menu will appear.

1 -> Interactive mode
2 -> Batch mode

Enter mode (0 = exit)

Type the number of the appropriate operation mode.

To exit the program at this point, type 0 (zero).

If Interactive mode is selected:

Enter part file name (<CR> = mode menu) :

Type the name of the part file without the **.pfm** extension.

To return to the mode menu, press <CR>.

Enter VDA file name (<CR> = <file_name>) :

Enter a name for the VDA file to be created.

To use the name displayed in parentheses, press <CR>.

If the specified file already exists, the following appears:

—- File <file_name>.igs already exists Replace existing file? (Y/N)

To write over the existing file, type **y**.

To enter a different name for the new file, type **n**.

Setting up part file <file_name> . . .
Translating entities from Cimatron to VDA:

NAME	POINT	ARC	CURVE	SURF	CONS	FACE
SET01	32	0	0	0	0	0
SET02	6	0	0	0	0	0
SET03	6	0	6	0	0	0
SET04	0	0	2	0	0	0
SET05	0	0	1	0	0	0
SET06	0	0	0	4	0	0
SET07	0	0	0	1	0	0
TOTAL	44	0	9	5	0	0

END OF WRITING THE VDA FILE

End of entities translation

Saving file <file name>.

Enter part file name (<CR> = mode menu)	Type the name of another part file to be converted OR Press <CR> to exit from the program. ❑

Chapter 20
The VDA Utilities Parameter File

The VDA Utilities parameter file allows the VDA utilities user to:

- Customize the interaction with the utilities.
- Insert data required by the utilities.

This chapter contains a general description of the creation, format and use of the parameter file. Further information on the **Read-VDA** parameters can be found in Chapter 18, **The Read-VDA Utility** chapter. Further information on the **Write-VDA** parameters can be found in Chapter 19, **The Write-VDA Utility** chapter.

A description of the **Parameter File Check** Utility is given at the end of this chapter.

Parameter File Name

The format of the parameter file name is **filename.vpr**. There are no limitations on the name, however the extension must be **.vpr**. For example, the name of the default parameter file is **default.vpr**.

Parameter File Creation

The **Cimatron** VDA parameter file is created by the user. Since the file is in ASCII (American Standard Code for Information Interchange) format, it can be easily created using any text editor. The **Cimatron** original default parameter file (**default.vpr**), located in the **<root_cad>\dat** directory, can be used as a template and guide for the user.

Parameter File Format and Rules

The parameter file is an ASCII file. The file contains statements which consist of predefined keywords and their parameter values. Following are some general rules of the parameter file format:

- The order of the lines is not important.
- Letter case is not important except when specifying **Cimatron** files, in which case the **Cimatron** file naming rules must be used.
- Each line starts with a keyword.
- If the same keyword appears more than once, and the statements can be accumulated, they will be; otherwise the last statement encountered will be used.
- A line that starts with ## is a comment line.
- Empty lines are ignored.
- Maximum of 50 words are allowed in one statement.
- To continue the arguments of a statement onto an additional line, finish the current line with \<CR> (backslash followed by carriage return) and continue on to the next line.
- The maximum number of characters allowed in one statement is 256.

Read-VDA Parameters

The parameter file (**.vpr**) contains parameters that control the operation mode, determine the interaction when running **Read-VDA** and provide information needed by the utility. These parameters will be described here.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP |<NOBACKUP>]}

  The parameter R-MODE selects the mode of operation. **Read-VDA** may be run interactively or in batch mode. Optional backup is available with batch mode.

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. User is prompted for name of each VDA file to be converted.
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the VDA files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.
BACKUP	Option used in batch mode. If a **.pfm** file already exists with the name specified in the batch file, it will be copied to a backup file with the extension **.bac.**
NOBACKUP	This is a system default option used in batch mode. If a part file (**.pfm**) with the name specified in the list already exists the new part file will replace it.

  Defaults: ASK, NOBACKUP

- SELECT R-SURFACES-AS {BEZIER | <VDA> | NURBS}

  Mode of reading VDA faces.

BEZIER	Surfaces are read as Bezier surfaces.
VDA	Surfaces are read as VDA surfaces.
NURBS	Surfaces are read as NURBS surfaces.

  Default : VDA

- SELECT R-CURVES-AS {<VDA> | NURBS}

  Mode of reading VDA curves.

  Default: VDA

- SELECT R-CONS-PREF-TYPE {<EQUAL> | PARAM | GEOM}

  Read-VDA-Preference Parameter for reading **Curve On Surface (CONS)** Entity.

  The parameter, R-CONS-PREF-TYPE, enables **Cimatron** users to define the preferred CONS representation to be read. Available representations are parametric and geometric.

EQUAL	Read both representations and sets **Cimatron** preferred flags to equally preferred.
PARAM	Read both representations and sets **Cimatron** preferred flags to parametric preferred.
GEOM	Read both representations and sets **Cimatron** preferred flags to geometric preferred.

  Default: EQUAL

- SELECT OPTION-MENU {OUT TO LOG | OUT TO SHELL}

  Output a Log File as an option.

OUT TO LOG	All translation data will be saved to a log file. The name of the log file will be that of the VDA file with the extension **.rvd**. The screen will display only information about the VDA type and form contained in the VDA file.
OUT TO SHELL	Translation data will be displayed on the screen.

  Default : OUT TO LOG

- SELECT R-GROUP-AS-LEVEL {YES|NO|<ASK>}

  Read group as level flag

YES	The group entities are read as level and set entities are ignored.
NO	Groups are ignored and set entities are read as level as before.
ASK	You are asked whether you want a read group.

- SELECT R-FIX-TRMSRF {NO | <VERIFY> | FIX}

  Mode of fixing trimmed surfaces.

  Default: VERIFY

- SELECT R-CONSTRUCT-FACE-FROM {<SRF. & BNDR.> | SURFACE | BOUNDARY}

  As the VDA format does not support a “slave” attribute, when translating a surface from VDA to a Cimatron PFM, the boundary curves of the surface are also translated like any other ordinary curve.

  This parameter enables you to choose which entities to translate.

SRF. & BNDR	The face and the boundary curves will be translated.
SURFACE	Only the face will be translated. The boundary curves will be deleted.
BOUNDARY	Only the boundary curves will be translated. No trimmed surfaces will be converted.

  Default: SRF. & BNDR

- SELECT R-SPLIT-SRF {<NO> | YES}

  Split a surface on each point when the surface is not smooth.

  This option checks every surface for smoothness. If a break point is found, the surface is split at the break point and two surfaces created as a result.

NO	Do not split on break points.
YES	Split surfaces on break points.

  Default: NO

- SELECT R-FIX-TRMSRF {NO| <VERIFY> | FIX}

  Mode of fixing trimmed surfaces.

  Default: VERIFY

- SELECT R-REFINE-TRMSRF-TOL {approximation-tolerance(0.0-1.0) | <0.0>}

  Refine trimmed surface boundaries when required (TOL > 0 and TOL < 1). The default option (0) parameter will not be used in the data interface.

  If the value is between 0 and 1, this will be used as the tolerance for the Refine Trmsrf algorithm which will be forced on the resulting trimmed surfaces after the data interface process is completed.

  If you set the value <0 or >1, the application will assume that 0.0 was used and the parameter will be ignored.

  Default: 0.0

Read-VDA Utility Interaction Statements

Select mode of operation:

SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP |<NOBACKUP>]

Read group as level flag

SELECT R-GROUP-AS-LEVEL {YES | NO | <ASK>}

Set the preference for reading a curve on a surface (cons) entity:

SELECT R-CONS-PREF-TYPE {<EQUAL> | PARAM | GEOM}

Set the preference for reading surfaces:

SELECT R-SURFACES-AS [BEZIER | <VDA> | NURBS}

Translate the face or the boundary curves or both of them:

SELECT R-CONSTRUCT-FACE-FROM {<SRF. & BNDR.> | SURFACE | BOUNDARY}

Write-VDA Parameters

The parameter file (**.vpr)** contains parameters that control the operation mode, determine the interaction when running **Write-VDA**, and provide information needed by the utility. These parameters will be described here.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT W-MODE {<ASK> | INTER | BATCH}

  The parameter W-MODE selects the mode of operation.

  Write-VDA may be run interactively (INTER) or in batch mode (BATCH).

  ASK Mode menu will appear. User is prompted for operation mode.

  INTER Program will run in interactive mode. The mode menu will not appear on the screen. The user is prompted for the name of each **Cimatron** part file to be converted.

  BATCH Program will run in batch mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.

  Default: ASK

- SELECT OPTION-MENU {OUT TO LOG | OUT TO SHELL}

  Output a Log File as an option.

  OUT TO LOG All translation data will be saved to a log file. The name of the log file will be that of the VDA file with the extension **.wvl**. The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

  OUT TO SHELL Translation data will be displayed on the screen.

  Default: OUT TO LOG

- SELECT W-TOL {approximation-tolerance (0.001-100) <0.1>}

  The parameter W-TOL sets the tolerance in mm used when approximating a **Cimatron** entity by a VDA entity.

  approximation tolerance

  User-defined tolerance; the range is 0.001 - 100 mm.

  Default: 0.1

- SELECT W-SOLID-EXPLODE-AS {SURFACES | BOUNDARIES | <BOTH>}

  Explode SOLID model to surfaces, boundaries, or both.

  Default: BOTH

Write-VDA Utility Interaction Statements

Select mode of operation:

```
SELECT W-MODE {<ASK> | INTER | BATCH}
```

Set tolerance, in mm, used when a **Cimatron** entity will be approximated by a VDA entity:

```
SET W-TOL {approximation-tolerance(0.001-100) <0.1>}
```

Explode SOLID model to surfaces, boundary or both.

```
SELECT W-SOLID-EXPLODE-AS {SURFACES | BOUNDARIES | <BOTH>}
```

USE FILE Statement - Nested Parameter File

This statement instructs the program to include a file which contains parameter statements during the processing of the parameter (**.vpr**) file. This enables the user to nest files of parameter statements.

The statement syntax is:

USE FILE {file-pathname}

- File-pathname is full absolute file name in the file system of the included file. There are no limitations on the name, extension or location of the included file.
- The included file is processed as soon as the USE FILE statement is encountered. The rule for multiple occurrences of the same statement is applied as if all processed statements from all nested files are in one parameter file.

As an example of how this feature can be used, consider a multiple user environment. Since certain information should be the same for each user, a system parameter file that defines all the global system parameters can be written. At the same time, certain information will be specific to each user (such as the author's name). To meet both needs, the user can create his own parameter file, which contains user-specific information, and also contains a USE FILE statement which includes the system parameter statements.

Default Parameter File and Parameter Values

The default parameter file, **default.vpr**, is located in one of the directories in the **Cimatron** file search path (the **Cimatron** file search rules are given at the beginning of this chapter). The original **Cimatron** default parameter file is located in the **<root_cad>\dat** directory. Authorization to modify this file should be restricted to the **Cimatron** system administrator.

Following is the default parameter file:

```
SELECT R-MODE ASK
SELECT R-LOG-FILE ASK
SELECT R-GROUP-AS-LEVEL ASK
SELECT R-CONS-PREF-TYPE EQUAL
SELECT R-SURFACES-AS-VDA VDA
SELECT R-CURVES-AS VDA
SELECT R-FIX-TRMSRF VERIFY
SELECT R-SPLIT-SRF NO
SELECT W-LOG-FILE ASK
SELECT W-SOLID-EXPLODE-AS BOTH
SET W-TOL 0.1
```

Parameter File Check

The VDA Utilities provides an option to check the format and statement syntax of the parameter file.

To check a file, use option **<3> PARAMETER FILE CHECK**, from the VDA Utilities menu.

If no errors are found in the parameter file, the following message will appear:

PARAMETER FILE WAS SUCCESSFULLY PARSED ❑

Section VI

JAMA-IS Utilities

Chapter 21
Introduction to the JAMA-IS Utilities

Part files, including geometric entities, dimensions and kinds of annotation, levels, views, masters and their instances, may all be transferred between **Cimatron** and other systems in the JAMA-IS format, approved by the Japan Automobile Manufacturers Association.

The following five functions are included in the **Cimatron** JAMA-IS Utilities:

Read-JAMA	Convert files that are in JAMA-IS format to files in **Cimatron** part file (**.pfm**) format.
Write-JAMA	Convert **Cimatron** part files (**.pfm**) to files in JAMA-IS format.
PARAMETER FILE CHECK	Verify a JAMA-IS parameter file (**.jpr**).
JAMA FILE INFORMATION	Generate a report (**.jrt**) on the JAMA-IS file tables and entities, and any syntax or structural errors in the file.
PFM FILE INFORMATION	Generate a report (**.pfr**) file on the **.pfm** file tables and entities, and any syntax or structural errors in the file.

Note:

- The interaction for each of these functions is identical to the corresponding IGES function interaction. For details, refer to the corresponding IGES chapters (Chapters 8 to 13) in this manual.

Running JAMA-IS

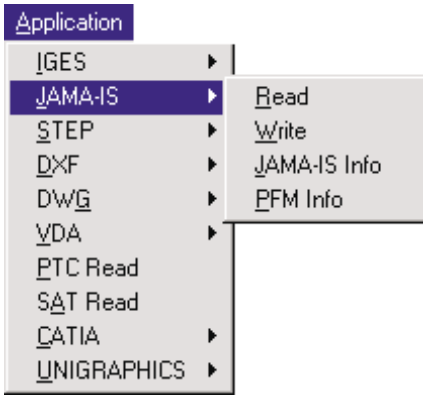
There are two methods of running the JAMA-IS Data Interface Utilities:

- **graphically**
- **textually**

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select JAMA-IS from the Application Menu. The following options are displayed:



3. Select the appropriate option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.

   For JAMA-IS, enter **iges -jama** at the DOS prompt.
2. The following JAMA-IS options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
4  ->  JAMA-IS  File  Information  option
5  ->  PFM  File  Information  option
```

3. Select the appropriate option.

Each of these utilities will be explained in a chapter of this section.

About the JAMA-IS Specification

"JAMA-IS" is an acronym for "Japan Automobile Manufacturers Association - IGES Subset" Specification. As its name implies, JAMA-IS is a subset of the IGES specification.

The necessity of forming this subset was due to problems with the IGES specification which prevents smooth data exchange. Some of these problems were manifested in different interpretation of the specification or translation accuracy in CAD/CAM systems.

To solve these problems, the Japan Automobile Manufacturers Association decided to form the JAMA-IS specification which deals with the special needs of the automotive industry. The main characteristics of the JAMA-IS specification are:

1. The limitation of the specification of geometric entities which are considered important to represent an automobile. Drawing and drafting entities have not been included in the specification due to the great difficulties and complexities of handling these entities. Although drafting entities are not part of the specification, they are considered as JAMA-IS acceptable entities and are exchanged according to the IGES specification.
2. Only the necessary and minimum entities were selected from the IGES specification. These entities are listed in the table below.
3. Each entity definition where its interpretation was considered to be ambiguous, was unified. This relates mainly to fields in the Directory Section and the Parameters Section.

Entities defined in JAMA-IS are as follows:

Type	Form	ENTITIES
100	-	Circular Arc
102	-	Composite Curve
104	1,2,3	Conic Arc
106	1,2,11,12	Copious Data
108	0	Plane
110	-	Line
112	-	Parametric Spline Curve
114	-	Parametric Spline Surface
116	-	Point
124	0	Transformation Matrix
126	0	Rational B-Spline Curve
128	0	Rational B-Spline Surface
142	-	Curve on a Parametric Surface
144	-	Trimmed (Parametric) Surface
308	-	Subfigure Definition
402	7	Associativity Instance (Group W/O Back Pointers)
404	0,1	Drawing
406	16	Property (Drawing Size)
408	-	Singular Subfigure Instance
410	0	View

Some other IGES entities are acceptable as JAMA-IS entities, whilst other are not. See the JAMA-IS Version 1.04 for details. Entities that are not defined as JAMA-IS entities, but are acceptable entities, should be exchanged according to the IGES version 5.0 specification. ❑

Chapter 22
JAMA-IS Options

JAMA-IS Utilities are invoked by one of the following methods:

- From the Data-Interface new interaction program (**cim90 -di**) selecting JAMA as application.
- From the DOS prompt, use **iges -jama** for textual interaction.

JAMA-IS Utilities are operated the same way as the equivalent IGES options, with the restriction of the JAMA-IS defined and acceptable entities.

JAMA-IS specific file types and their unique extensions:

.rjl	Read-JAMA Log File
.wjl	Write-JAMA Log File
.jpr	JAMA-IS Parameter File
.jrt	JAMA-IS File Information

The differences in entity or attribute translation between Write-IGES and Write-JAMA, are as follows:

- The **Cimatron** Offset Curve is approximated to the JAMA-IS Parametric Spline Curve (Type 112).
- All **Cimatron** surfaces, with the exception of IGES and NURBS surfaces, are approximated/converted to JAMA-IS Parametric Spline Surface (Type 114).
- **Cimatron** Planar Face (Class 4, Type 1) is converted to JAMA-IS Trimmed Surface (Type 144).
- The **Cimatron** Color attribute is converted to the nearest JAMA-IS pre-defined color.

JAMA-IS Parameters File is based on IGES Parameters File with the following exceptions:

- W-SURFACE-AS-114 and W-SURFACE-AS-128 are not available in the JAMA-IS Parameter File, as all the surfaces, with the exception of the **Cimatron** NURBS surface, are approximated to JAMA-IS Parameter Spline Surfaces (Type 114).
- W-COLOR-ENTITY is not available as all the color attributes are converted to a JAMA-IS pre-defined color.

JAMA-IS File Structure

```
JAMA-IS VERSION  1.0                                                    S    1
Cimatron Jama-Is convertor                                              S    2
1H,,1H;,4HNONE,27HE:\UDD\USER\test\angle2.igs,14HCimatron V9.0,43HCimat G    1
ron JAMA-IS (V1.0) Convertor V1.0.6.47,16,24,8,56,8,4HNONE,1.00000,2,2HM G    2
M,8,0.400000,13H961126.092804,0.000100000,10000.0,4HUser,8HCimatron,8,0,13H9 G    3
61126.092724;                                                           G    4
212      1      0      1      0      0      0                      000010100D    1
212      1      8      2      0                                            0D    2
110      3      0      1      0      0      0                      000010100D    3
110      1      5      2      0                                            0D    4
212      5      0      1      0      0      0                      000010100D    5
212      1      2      2      0                                            0D    6
212      7      0      1      0      0      0                      000010100D    7
212      1      7      2      0                                            0D    8
212      9      0      1      0      0      0                      000010100D    9
212      1      6      2      0                                            0D   10
212     11      0      1      0      0      0                      000010100D   11
212      1      5      2      0                                            0D   12
110     13      0      1      0      0      0                      000010100D   13
110      1      6      2      0                                            0D   14
110     15      0      1      0      0      0                      000010100D   15
110      1      8      2      0                                            0D   16
110     17      0      1      0      0      0                      000010100D   17
110      1      2      2      0                                            0D   18
110     19      0      1      0      0      0                      000010100D   19
110      1      7      2      0                                            0D   20
404     21      0      0      0      0      0                     000000100D   21
404      0      0      1      0                                            0D   22
212,1,8,61.2541,10.23737,1,1.570796,0.,0,0,-6.923902,-17.59825,           1P    1
0.,8Hangale=0,0;                                                         1P    2
110,-60.260904673567,39.152565183391,0.,-60.260904673567,                3P    3
101.15256518339,0.,0;                                                    3P    4
212,1,8,61.2541,10.24738,1,1.570796,.5235988,0,0,-28.26035,              5P    5
25.02218,0.,8Hangle=30,0;                                                5P    6
212,1,9,68.05399,10.25738,1,1.570796,3.141593,0,0,-109.297,              7P    7
-20.83015,0.,9Hangle=180,0;                                              7P    8
212,1,9,70.51577,10.23737,1,1.570796,4.18879,0,0,-88.01922,              9P    9
-52.3994,0.,9Hangle=240,0;                                               9P   10
212,1,8,61.16404,10.24237,1,1.570796,1.570796,0,0,-67.11833,             11P   11
39.74112,0.,8Hangle=90,0;                                               11P   12
110,-94.30661352869,-50.728106194133,0.,-128.30661352869,               13P   13
-109.61783365147,0.,0;                                                  13P   14
110,-5.0956942681928,-23.7799065849,0.,54.904305731807,                 15P   15
-23.7799065849,0.,0;                                                    15P   16
110,-24.119619536112,20.722490023984,0.,29.573955498523,                17P   17
51.722490023984,0.,0;                                                   17P   18
110,-110.40670914418,-15.287082804578,0.,-178.40670914418,              19P   19
```

Figure 22-1: Sample ASCII JAMA-IS File

❑

Section VII

STEP Utilities

Chapter 23
Introduction to the STEP Utility

The **Cimatron STEP Utility** enables you to transfer geometric entities, in the **STEP** format, between Cimatron and other software packages.

The following two functions are included in the Cimatron STEP Utilities:

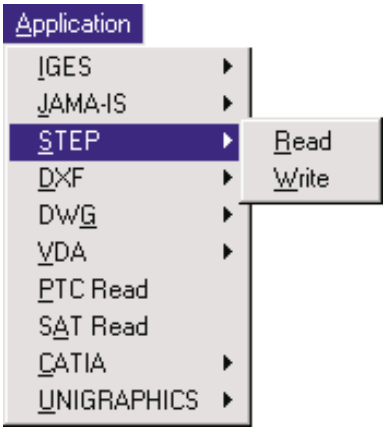
Read-STEP	Convert files from the STEP AP203 and AP214 formats to files in Cimatron part file (**.pfm**) format.
Write-STEP	Convert Cimatron part files (**.pfm**) to files in the STEP AP203 format.

Running STEP

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select STEP from the Application Menu. The following options are displayed:



About STEP

STEP is being defined by approximately 300 experts from 15 countries. This effort began in 1983 when the ISO (International Standards Organization) formed the Technical Committee on Industrial Automation Systems and Integration.

The STEP standard has the designation ISO 10303.

The **Cimatron** STEP Utility currently provides a STEP Read ability for STEP files in the ISO AP203 format.

What is STEP?

CAD/CAM systems are in use today in increasing numbers for applications covering all phases of product's design, analysis, manufacturing and testing. As users in each of these stages may use systems supplied by different vendors, there is a need for data to be exchanged digitally between these systems.

The STEP initiative is an international effort to standardize the exchange of product design and manufacturing data. STEP is an acronym for Standard for Exchange of Product model data. Within the United States, it is also known as PDES (Product Data Exchange using STEP).

STEP File Structure

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION ((‘NIST ParaSolid Test Parts’), ‘1’);
FILE_NAME (‘ PRT0001’, ‘1996-12-23T13:12:24-05:00’, (Edi Fleishman), (‘NIST’),
 ‘ST-DEVELOPER v1.3’, ‘EDS - UNIGRAPHICS 10.5’, ‘’);
FILE_SCHEMA ((‘CONFIG_CONTROL_DESIGN’));
ENDSEC;
DATA;
#1=DIRECTION(‘’,(1.E0,0.E0,0.E0));
#2=VECTOR(‘’,#1,8.08E0);
#3=CARTESIAN_POINT(‘’,(-8.08E0,0.E0,-3.42E0));
#4=LINE(‘’,#3,#2);
#5=TRIMMED_CURVE(‘’,#4,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#6=GEOMETRIC_CURVE_SET(‘’,(#5,#11,#16,#21,#26,#31,#36,#41,#46,#51,#56,#61));
#7=DIRECTION(‘’,(0.E0,1.E0,0.E0));
#8=VECTOR(‘’,#7,1.104E1);
#9=CARTESIAN_POINT(‘’,(0.E0,0.E0,-3.42E0));
#10=LINE(‘’,#9,#8);
#11=TRIMMED_CURVE(‘’,#10,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#12=DIRECTION(‘’,(-1.E0,0.E0,0.E0));
#13=VECTOR(‘’,#12,8.08E0);
#14=CARTESIAN_POINT(‘’,(0.E0,1.104E1,-3.42E0));
#15=LINE(‘’,#14,#13);
#16=TRIMMED_CURVE(‘’,#15,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#17=DIRECTION(‘’,(0.E0,-1.E0,0.E0));
#18=VECTOR(‘’,#17,1.104E1);
#19=CARTESIAN_POINT(‘’,(-8.08E0,1.104E1,-3.42E0));
#20=LINE(‘’,#19,#18);
#21=TRIMMED_CURVE(‘’,#20,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#22=DIRECTION(‘’,(1.E0,0.E0,0.E0));
#23=VECTOR(‘’,#22,8.08E0);
#24=CARTESIAN_POINT(‘’,(-8.08E0,0.E0,0.E0));
#25=LINE(‘’,#24,#23);
#26=TRIMMED_CURVE(‘’,#25,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#27=DIRECTION(‘’,(0.E0,-1.E0,0.E0));
#28=VECTOR(‘’,#27,1.104E1);
#29=CARTESIAN_POINT(‘’,(-8.08E0,1.104E1,0.E0));
#30=LINE(‘’,#29,#28);
#31=TRIMMED_CURVE(‘’,#30,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#32=DIRECTION(‘’,(-1.E0,0.E0,0.E0));
#33=VECTOR(‘’,#32,8.08E0);
```

```
#34=CARTESIAN_POINT('',(0.E0,1.104E1,0.E0));
#35=LINE('',#34,#33);
#36=TRIMMED_CURVE('',#35,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#37=DIRECTION('',(0.E0,1.E0,0.E0));
#38=VECTOR('',#37,1.104E1);
#39=CARTESIAN_POINT('',(0.E0,0.E0,0.E0));
#40=LINE('',#39,#38);
#41=TRIMMED_CURVE('',#40,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#42=DIRECTION('',(0.E0,0.E0,1.E0));
#43=VECTOR('',#42,3.42E0);
#44=CARTESIAN_POINT('',(-8.08E0,0.E0,-3.42E0));
#45=LINE('',#44,#43);
#46=TRIMMED_CURVE('',#45,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#47=DIRECTION('',(0.E0,0.E0,1.E0));
#48=VECTOR('',#47,3.42E0);
#49=CARTESIAN_POINT('',(0.E0,0.E0,-3.42E0));
#50=LINE('',#49,#48);
#51=TRIMMED_CURVE('',#50,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#52=DIRECTION('',(0.E0,0.E0,1.E0));
#53=VECTOR('',#52,3.42E0);
#54=CARTESIAN_POINT('',(0.E0,1.104E1,-3.42E0));
#55=LINE('',#54,#53);
#56=TRIMMED_CURVE('',#55,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#57=DIRECTION('',(0.E0,0.E0,1.E0));
#58=VECTOR('',#57,3.42E0);
#59=CARTESIAN_POINT('',(-8.08E0,1.104E1,-3.42E0));
#60=LINE('',#59,#58);
#61=TRIMMED_CURVE('',#60,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
#62=DIMENSIONAL_EXPONENTS(1.E0,0.E0,0.E0,0.E0,0.E0,0.E0,0.E0);
#63=(LENGTH_UNIT()NAMED_UNIT(#62)SI_UNIT(.MILLI.,.METRE.));
#64=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(2.54E1),#63);
#65=(CONVERSION_BASED_UNIT('INCH',#64)LENGTH_UNIT()NAMED_UNIT(#62));
#66=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#67=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
#68=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNIT_ASSIGNED_CONTEXT((#65,#66,#67) REPRESENTATIO
N_CONTEXT('ID1','3'));
#69=GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION('',(#6),#68);
#70=APPLICATION_CONTEXT ('CONFIGURATION CONTROLLED 3D DESIGNS OF MECHANICAL PARTS AND ASSEMBLIES
');
#71=APPLICATION_PROTOCOL_DEFINITION('international standard', 'config_control_design',1994,#70);
#72=DESIGN_CONTEXT('',#70,'design');
#73=MECHANICAL_CONTEXT('',#70,'mechanical');
#74=PRODUCT('PRT0001','PRT0001','NOT SPECIFIED',(#73));
#75=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE('','LAST_VERSION',#74,.MADE.);
#76=PRODUCT_DEFINITION('design','',#75,#72);
#77=PRODUCT_DEFINITION_SHAPE('','SHAPE FOR PRT0001.',#76);
#78=SHAPE_DEFINITION_REPRESENTATION(#77,#69);
#79=PRODUCT_CATEGORY('part','');
#80=PRODUCT_RELATED_PRODUCT_CATEGORY('detail','',(#74));
#81=PRODUCT_CATEGORY_RELATIONSHIP('','',#79,#80);
#82=SECURITY_CLASSIFICATION_LEVEL('unclassified');
#83=SECURITY_CLASSIFICATION('','',#82);
#84=CC_DESIGN_SECURITY_CLASSIFICATION(#83,(#75));
#85=APPROVAL_STATUS('approved');
#86=APPROVAL(#85,'');
#87=CC_DESIGN_APPROVAL(#86,(#83,#75,#76));
#88=CALENDAR_DATE(96,23,12);
#89=COORDINATED_UNIVERSAL_TIME_OFFSET(0,0,.BEHIND.);
#90=LOCAL_TIME(13,12,0.E0,#89);
#91=DATE_AND_TIME(#88,#90);
#92=APPROVAL_DATE_TIME(#91,#86);
#93=DATE_TIME_ROLE('creation_date');
#94=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#91,#93,(#76));
#95=DATE_TIME_ROLE('classification_date');
#96=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#91,#95,(#83));
#97=PERSON('UNSPECIFIED','UNSPECIFIED',$,$,$,$);
#98=ORGANIZATION('UNSPECIFIED','UNSPECIFIED','UNSPECIFIED');
#99=PERSON_AND_ORGANIZATION(#97,#98);
#100=APPROVAL_ROLE('approver');
#101=APPROVAL_PERSON_ORGANIZATION(#99,#86,#100);
#102=PERSON_AND_ORGANIZATION_ROLE('creator');
#103=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#99,#102,(#75,#76));
#104=PERSON_AND_ORGANIZATION_ROLE('design_supplier');
#105=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#99,#104,(#75));
#106=PERSON_AND_ORGANIZATION_ROLE('classification_officer');
#107=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#99,#106,(#83));
#108=PERSON_AND_ORGANIZATION_ROLE('design_owner');
#109=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#99,#108,(#74));
ENDSEC;
END-ISO-10303-21;
```

Header Section

The section contains file description, file name and file schema.

```
HEADER;

FILE_DESCRIPTION (('NIST ParaSolid Test Parts'), '1');
```

description	informal description of contents	'NIST ParaSolid Test Parts'
implementation_level	required implementation level of post-processor.	'1'

```
FILE_NAME ('design1', '1996-12-23T13:12:24-05:00', (Edi  Fleishman), ('NIST'),
 'ST-DEVELOPER v1.3', 'EDS - UNIGRAPHICS 10.5', '');
```

name	name of exchange file.	'PRT0001'
time_stamp	file creation and date	'1996-12-23 T13:12:24-05:00'

author	person who created the exchange file	Edi Fleishman
organization	organization with which author is identified.	NIST
preprocessor_version	identification of software used to create the file (name and version)	ST-DEVELOPER v1.3
originating_system	the system from which the data originated	'EDS - UNIGRAPHICS 10.5'
authorization	who authorized the sending of the exchange file.	unspecified

```
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
```

schema_identifiers	identifies schema which specify the instance in the data section of the file (usually the single name of longform of the application interpreted model.	'CONFIG_CONTROL_DESIGN

```
ENDSEC;
```

Data Section

The data comprises a list of entity_occurrences. Each entity occurrence has a unique identifier and is an instance of a type specified in the application protocol, of which the file as a whole is an instance.

```
DATA;
```

Entities representation description:

```
#69=GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION('',(#6),#68);
```

Entities set:

```
#6=GEOMETRIC_CURVE_SET('',(#5,#11,#16,#21,#26,#31,#36,#41,#46,#51,#56,#61));
```

First curve:

```
#5=TRIMMED_CURVE('',#4,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
```

Curve type

```
#4=LINE('',#3,#2);
```

Origin point

```
#3=CARTESIAN_POINT('',(-8.08E0,0.E0,-3.42E0));
```

Vector & size

```
#2=VECTOR('',#1,8.08E0);
```

Vector direction

```
#1=DIRECTION('',(1.E0,0.E0,0.E0));
```

Second curve:

```
#11=TRIMMED_CURVE('',#10,(0.E0),(1.E0),.T.,.UNSPECIFIED.);
ENDSEC;
END-ISO-10303-21; ❑
```

Chapter 24
The Read-STEP Utility

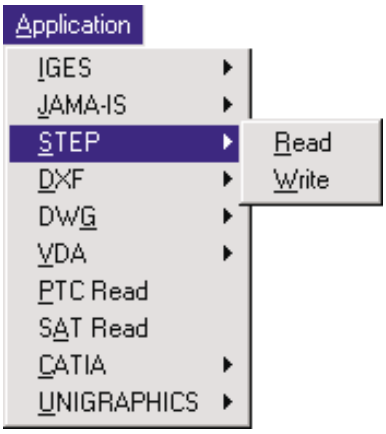
The **Read-STEP** Utility converts geometric entities from the STEP AP203 and AP214 formats to files in the **Cimatron** part file (**.pfm**) format.

Running Read-STEP

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select STEP from the Application Menu. The following options are displayed:



3. Select the **Step, Read** option.

Textual

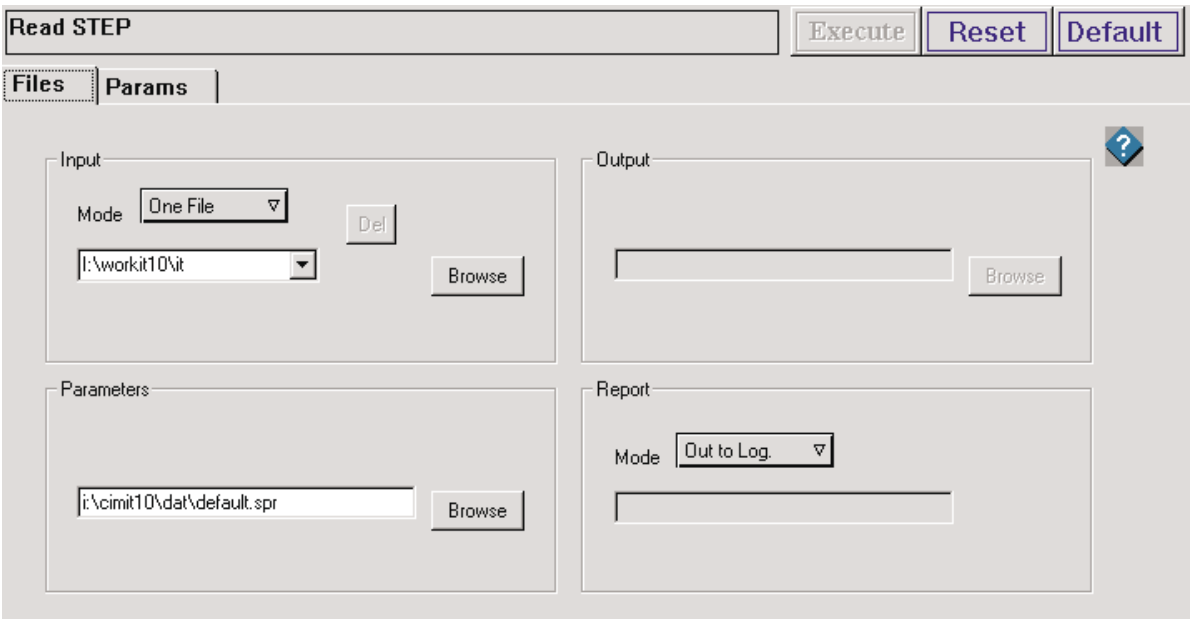
How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. The following STEP options are available:

```
1  ->  Read  option
2  ->  Write  option
```

3. Select the appropriate option.

Graphic Interaction with Read-STEP

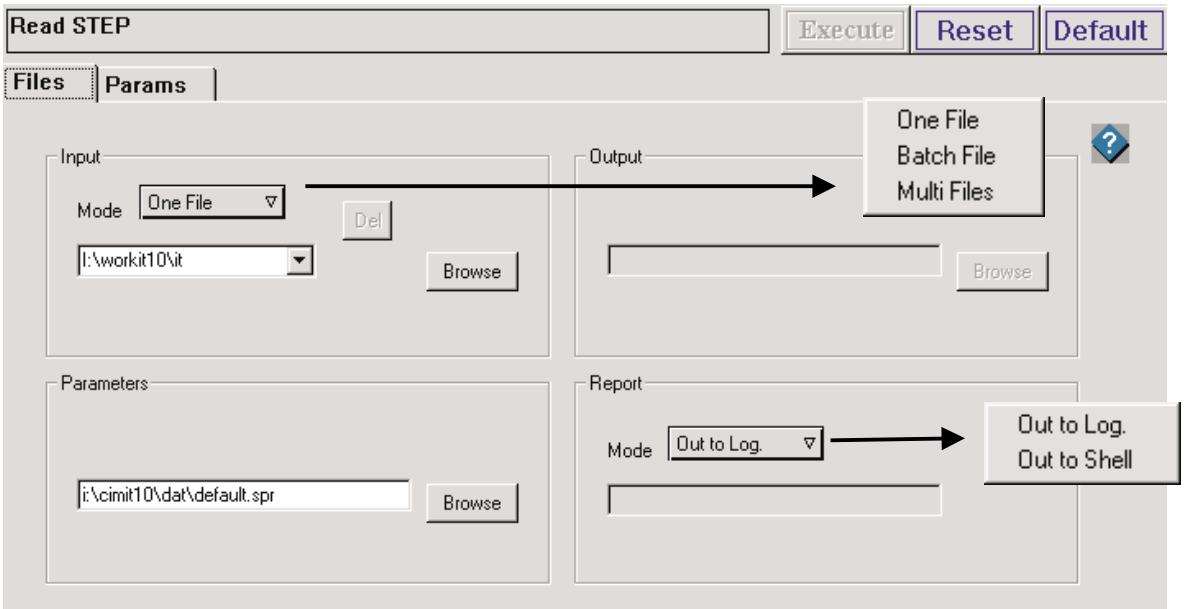


How To:

1. Set the parameters.
2. Select a file in the Input area using the Browse button.
3. Press the Execute button.

Files

This dialog defines the Input and Output files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input **STEP** file (the file to be converted to a Cimatron **.pfm**).

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.pfm** file is identical to the prefix of the input STEP file.

   Choose a different **.pfm** file, or using the Browse button, select an existing **.pfm** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the STEP parameter file is **default.spr**. You may use this name or select an existing parameter file name using the Browse button.

The parameter file enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read STEP translation data **svl** file is identical to the prefix of the input **.stp** file.

   Choose a different **.svl** file, or using the Browse button, select an existing **.svl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

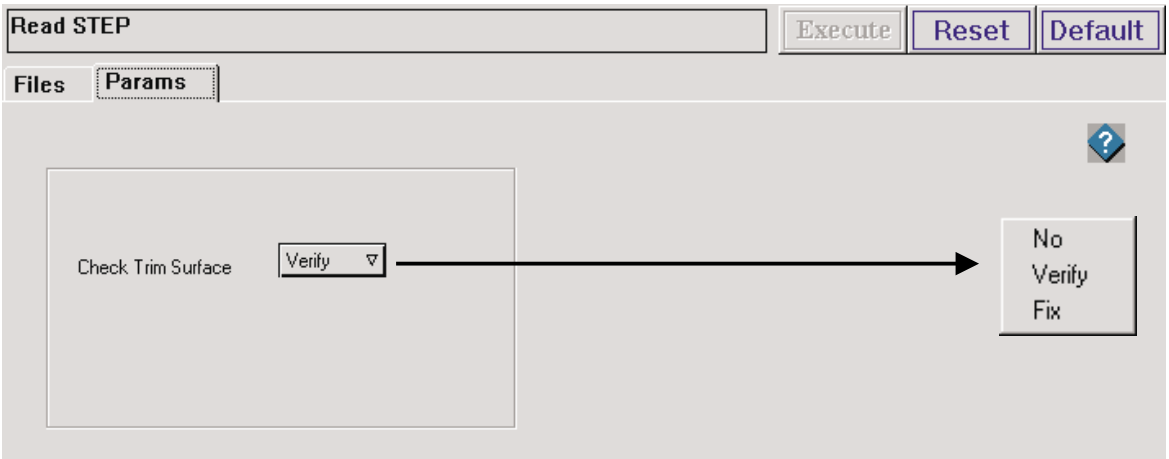
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **Check Trim Surface**: Confirm whether to check trimmed surfaces: No, Verify of Fix.

Entity Mapping

Table 24-1 Mapping of STEP Entities to Cimatron

STEP Entity	Cimatron Entity
ADVANCED_BREP_SHAPE_REPRESENTATION	Trimmed Surfaces
ADVANCED_FACE	Trimmed Surfaces
AXIS2_PLACEMENT_3D	
BEZIER_CURVE	Bezier spline curve
BEZIER_SURFACE	Bezier spline surface
BOUNDARY_CURVE	
BOUNDED_CURVE	
BOUNDED_SURFACE	Trimmed Surfaces
BOUNDED_SURFACE_CURVE	
BREP_WITH_VOIDS	Trimmed Surfaces
B_SPLINE_CURVE	NURBS curve
B_SPLINE_CURVE_WITH_KNOTS	NURBS curve
B_SPLINE_SURFACE	NURBS surface
B_SPLINE_SURFACE_WITH_KNOTS	NURBS surface
CARTESIAN_POINT	Point
CARTESIAN_TRANSFORMATION_OPERATOR	
CARTESIAN_TRANSFORMATION_OPERATOR_2D	
CARTESIAN_TRANSFORMATION_OPERATOR_3D	
CIRCLE	NURBS
CLOSED_SHELL	Trimmed Surfaces
COMPOSITE_CURVE	NURBS Curve
COMPOSITE_CURVE_SEGMENT	NURBS Curve
CONIC	NURBS Curve
CONICAL_SURFACE	NURBS Curve
CURVE	NURBS Curve
CURVE_BOUNDED_SURFACE	Trimmed Surfaces
CYLINDRICAL_SURFACE	NURBS Surface
EDGE	
EDGE_CURVE	
EDGE_LOOP	
ELLIPSE	NURBS curve
FACE	Trimmed Surfaces
FACE_BOUND	
FACE_OUTER_BOUND	
FACE_SURFACE	Trimmed Surfaces
GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION	Trimmed Surfaces
GEOMETRIC_CURVE_SET	
GEOMETRIC_SET	
HYPERBOLA	NURBS curve
INTERSECTION_CURVE	NURBS curve
LOOP	
MANIFOLD_SOLID_BREP	Trimmed Surfaces
MANIFOLD_SURFACE_SHAPE_REPRESENTATION	Trimmed Surfaces
MAPPED_ITEM	
NEXT_ASSEMBLY_USAGE_OCCURRENCE	
OPEN_SHELL	Trimmed Surfaces

STEP Entity	Cimatron Entity
ORIENTED_CLOSED_SHELL	Trimmed Surfaces
ORIENTED_EDGE	
ORIENTED_FACE	Trimmed Surfaces
ORIENTED_OPEN_SHELL	Trimmed Surfaces
OUTER_BOUNDARY_CURVE	
PARABOLA	NURBS curve
PLANE	NURBS Surfaces
POINT	Point
QUASI_UNIFORM_CURVE	NURBS curve
QUASI_UNIFORM_SURFACE	NURBS Surfaces
RATIONAL_B_SPLINE_CURVE	NURBS curve
RATIONAL_B_SPLINE_SURFACE	NURBS Surfaces
RECTANGULAR_TRIMMED_SURFACE	NURBS Surfaces
SEAM_CURVE	NURBS curve
SHELL_BASED_SURFACE_MODEL	Trimmed Surfaces
SPHERICAL_SURFACE	NURBS Surfaces
SURFACE	NURBS Surfaces
SURFACE_CURVE	NURBS curve
SURFACE_OF_LINEAR_EXTRUSION	NURBS Surfaces
SURFACE_OF_REVOLUTION	NURBS Surfaces
TOROIDAL_SURFACE	NURBS Surfaces
TRIMMED_CURVE	NURBS curve
UNIFORM_CURVE	NURBS curve
UNIFORM_SURFACE	NURBS Surfaces
VECTOR	
VERTEX	
VERTEX_POINT	

❑

Chapter 25
The Write-STEP Utility

The Write-STEP Utility converts geometric entities from Cimatron part files (.pfm) to files in the STEP AP203 or AP214 format.

Running Write-STEP

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select STEP from the Application Menu. The following options are displayed:
3. Select the Step, Write option.

Textual

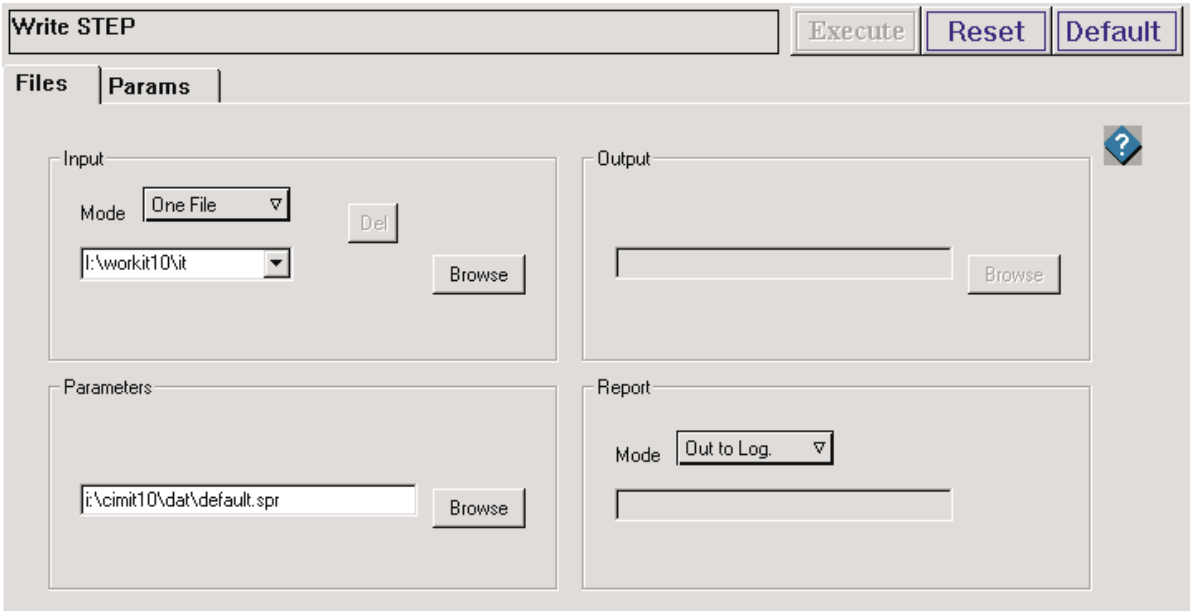
How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. The following STEP options are available:

   1 -> Read option
   2 -> Write option

3. Select the appropriate option.

Graphic Interaction with Write-STEP

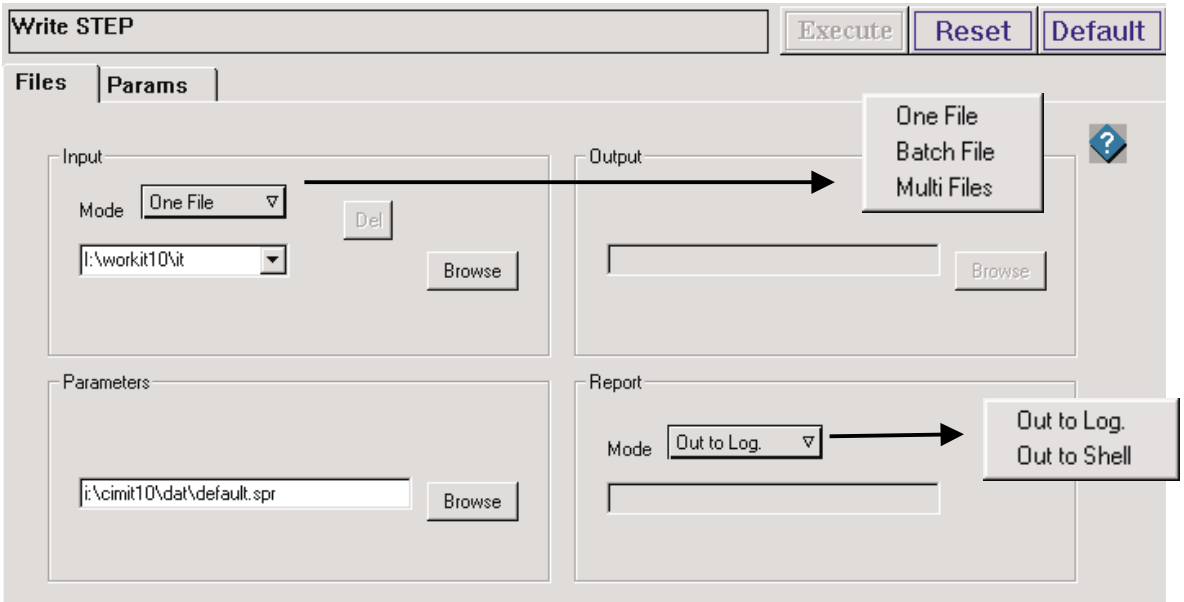


How To:

1. Set the parameters.
2. Select a file in the Input area using the Browse button.
3. Press the Execute button.

Files

This dialog defines the Input and Output files associated with the translation.



Input

The Read/Write operation may be run in either interactive or batch mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input STEP file (the file to be converted to a Cimatron .pfm).

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output .pfm file is identical to the prefix of the input STEP file.

   Choose a different .pfm file, or using the Browse button, select an existing .pfm file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the STEP parameter file is **default.spr**. You may use this name or select an existing parameter file name using the Browse button.

The parameter file enables you to:

- Customize the interaction with the data interface Read/Write.
- Insert data required by the data interface Read/Write.
- Influence the behavior of the data interface Read/Write by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read STEP translation data **svl** file is identical to the prefix of the input **.stp** file.

   Choose a different **.svl** file, or using the Browse button, select an existing **.svl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

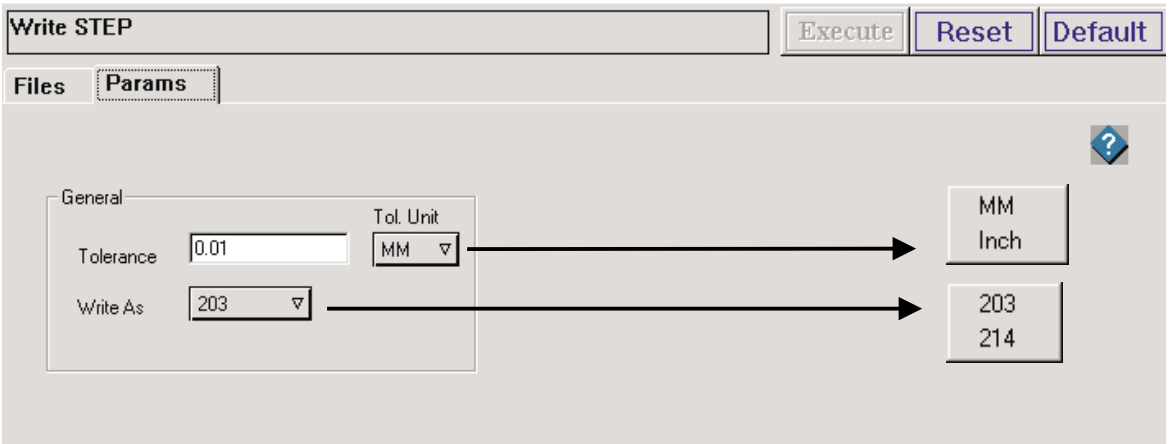
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **Tolerance:** Set the tolerance in MM when approximating a Cimatron entity by an entity in another standard. The tolerance range is 0.001 - 100. The default is 0.1.
2. **Tol. Unit**: Set the unit of tolerance in the Tolerance field.
3. **Write As:** Convert Cimatron It entities into the STEP AP203 or AP214 format.

Entity Mapping

Table 25-1 Mapping of Cimatron Entities to STEP

Cimatron Entities	STEP Entity
Circle	Rational_B_Spline_Curve
Composite Curve	Rational_B_Spline_Curve Or B_Spline_Curve_With_Knots
Conic	Rational_B_Spline_Curve
2D String/3D string	B_Spline_Curve_With_Knots
Planar Face	Open_Shell
Line	B_Spline_Curve_With_Knots
NURBS Curve	B_Spline_Curve_With_Knots
NURBS Surface	B_Spline_Surface_With_Knots
Ruled Surface	Rational_B_Spline_Surface
Surface of Revolution	Rational_B_Spline_Surface
Parallel Drive Surface	Rational_B_Spline_Surface Or B_Spline_Surface_With_Knots
NURBS Curve	Rational_B_Spline_Curve Or B_Spline_Curve_With_Knots
NURBS Surface	Rational_B_Spline_Surface Or B_Spline_Surface_With_Knots
Trimmed Surface	Open_Shell

❑

Section VIII

SAT Utilities

Chapter 26
Introduction to the SAT Utility

About SAT

The **Cimatron** SAT Utility inputs a file in the SAT format (**.sat**) and converts it to a **Cimatron** solid part file (**.pfm**).

The following utility is included in the Cimatron SAT Utilities:

Read-SAT	Convert files that are in the SAT format to files in Cimatron format. The program tests the **.sat** file for ACIS validity.

Running SAT

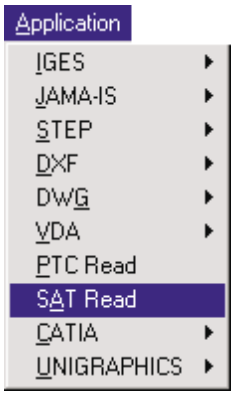
There are two methods of running the SAT Utility:

- **graphically**
- **textually**

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select SAT Read from the Application Menu:



Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following SAT options are available:

```
1 -> Read option
```

3. Select the appropriate option.

This option will be explained in the following chapter. ❑

Chapter 27
The Read-SAT Utility

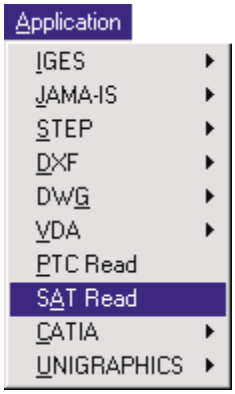
The Read-SAT Utility converts files in the SAT format to files in the **Cimatron** solid part file (**.pfm**) format.

Running Read-SAT

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select SAT Read from the Application Menu:



Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following SAT options are available:

```
1 -> Read option
```

3. Select the **Read** option.

Running Modes

The **Read-SAT** Utility may be run interactively, or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

*Or*In **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file as containing one file name per line.

Notes:

- Each part file created will be in the same directory and have the same name as the SAT file, but will have the extension **.pfm** instead of **.sat**.
- The SAT file may be located in the current or other directories. Enter the full path name of the SAT file if it is not located in the current directory.
- In **Batch Mode Without Backup**, if a file already exists with the same name, including the .**pfm** extension, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and **.pfm** files already exist with the name specified in the list, the old **.pfm** file will be copied to a backup file with the extension **.bac**. Then a new **.pfm** file will be created from the sat file.

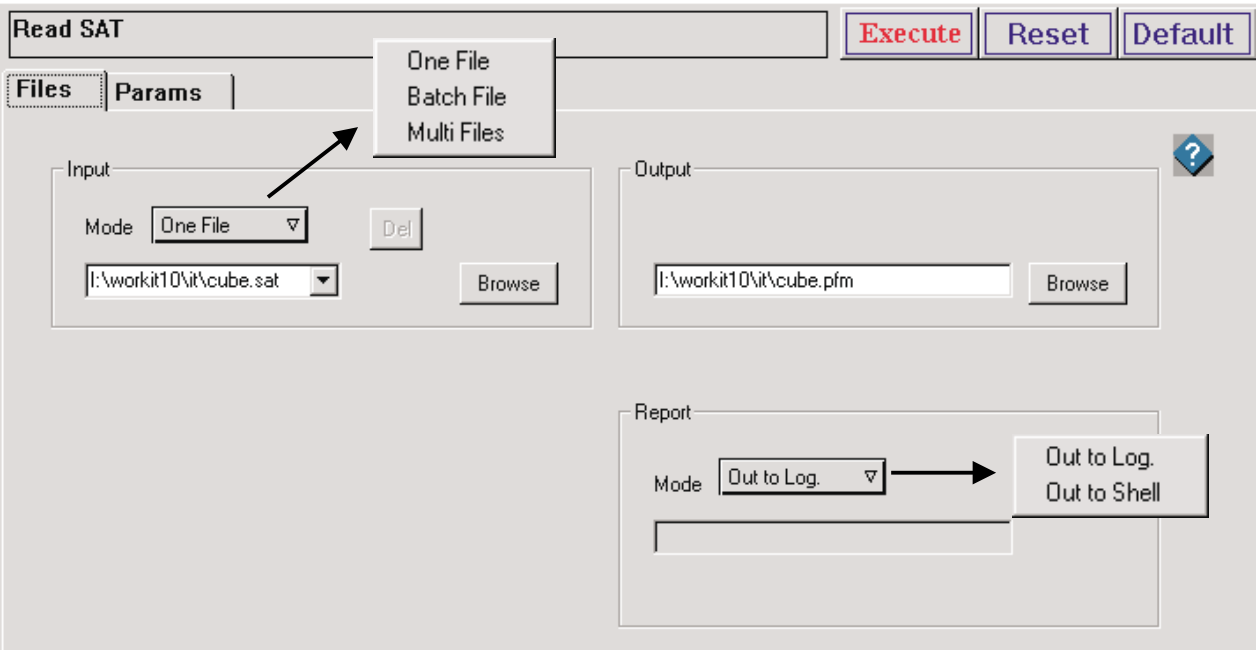
Graphic Interaction with Read-SAT

How To:

1. Set the parameters.
2. Select a file in the Input area using the Browse button.
3. Press the Execute button.

Files

This dialog defines the Input and Output files associated with the translation.



Input

The Read/Write operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input SAT file (the file to be converted to a Cimatron **.pfm**).
3. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.

The available Input Mode options are:

- ONE FILE — Run the Read/Write operation interactively, one file at a time.
- BATCH FILE — Run the Read/Write operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option.
- MULTI FILES — Run the Read/Write operation interactively, multiple files at a time.
- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Notes:

- We recommended that you view the SAT files that did not pass the Acis Test.

  To do so you can download the free Acis Viewer from www.spatial.com.
- The shade may sometimes be problematic due to display tolerance. It is recommended to set the display tolerance in solid-util-set tolerance -display tolerance, to a lower value.
- An error message “EXECUTION OF FEATURE ACIS SOL failed” indicates that the pfm file is empty.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- A listing of each entity translated and any errors or problems encountered during translation.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read STEP translation data **svl** file is identical to the prefix of the input **.stp** file.

   Choose a different **.svl** file, or using the Browse button, select an existing **.svl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

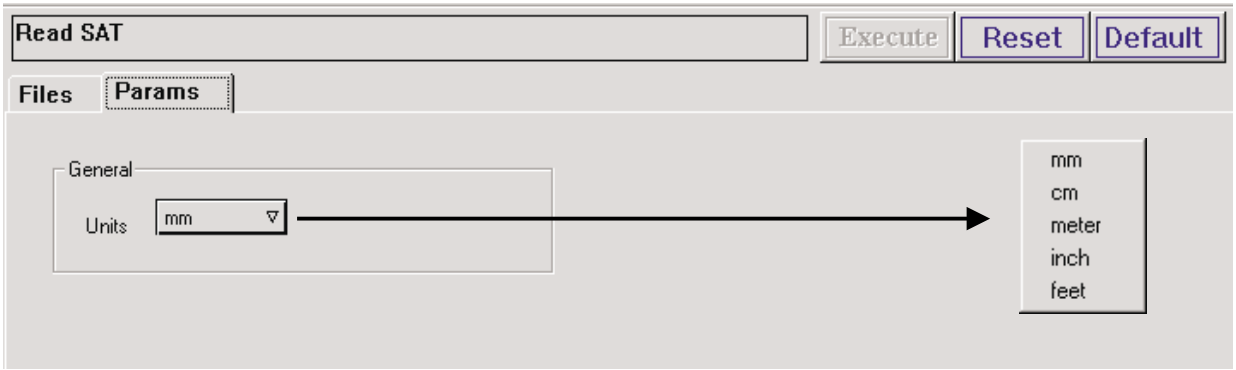
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **Unit**: Select the units of measurement of the **pfm** file: mm, cm, meter, inch or feet.

   If the **sat** file has a unit parameter in the file header, the **pfm** file will be in the same unit. If the unit of the source is not defined, use this option to define the unit of the **pfm** file.

Textual Interaction with Read-SAT

The following section describes the interaction upon invoking **Read-SAT**.

Cimatron (version 11.0) SAT (version 5.0) Utilities (version number)
==========

Available options:
1 - Read option
Enter option = exit

SAT (version 5.0) - Cimatron (version 11.0) Converter (version number)

==========

Available modes:	The mode menu will appear.
1 -> Interactive mode *2 -> Batch mode with backup* *3 -> Batch mode without backup*	
Enter mode (0 = exit)	Type the number of the appropriate operation mode. To exit the program at this point, type 0 (zero).

If **Interactive mode** is selected:

Enter SAT file name (<CR>=mode menu)	Type the name of the SAT file to be converted. To exit the program at this point, press <CR>.
Enter part file name (<CR>=<file_name>) :	Type a name for the part file. To use the name displayed, press <CR>. If the specified file already exists, the following message appears:

—- File <file_name>.pfm already exists

Replace existing file ? (Y/N)	To write over the existing file, type **y**. To type a different name for the file, type **n**.

SAT (version 5.0) - Cimatron (version 11.0) Converter (version number)

==========

checked:	A list of entities that was checked for ACIS validity.
Setting up part file file_name ...	The new part file will be saved with the **.sat** extension.

Translating entities from SAT to Ciatron :
End of entities translation
Saving part <file_name>

❑

Section IX

CATIA Utilities

Chapter 28
Introduction to the CATIA Utilities

The **Cimatron** Read-CATIA Utility inputs a model file created by CATIA versions 3 and 4 and converts it to a **Cimatron** part file (**.pfm**).

The **Cimatron** Read-CATIA Utility is an independent utility that does not require a CATIA license. It will only handle geometric CATIA data form.

Using the **Cimatron** Read-CATIA utility, the following files can be converted to **Cimatron** part file (**.pfm**) format:

without any extension
.model
.mdl
.cat
.mod
.dlv3
.dlv4
.dlv
.exp

The Cimatron Read-CATIA Utility automatically recognizes the CATIA native file by checking its header, irrespective of the file extension.

The following two functions are included in the **Cimatron** CATIA Utilities:

Read-CATIA	Convert files that are in CATIA format to files in **Cimatron** part file (**.pfm**) format.
Write-CATIA	Convert **Cimatron** part files (**.pfm**) to files in the CATIA format.
PARAMETER FILE CHECK	Verify a CATIA parameter file (**.cpr**).

Running CATIA

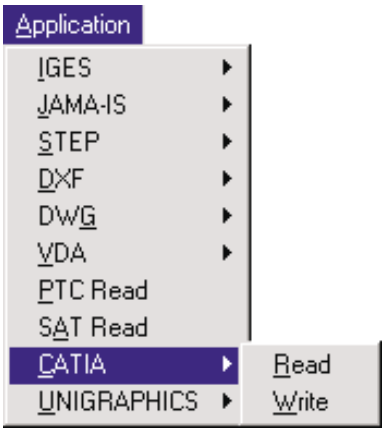
There are two methods of running the CATIA Utility:

- **graphically**, using the usual Windows graphical user interfaces,
- **textually**, entering commands at the DOS prompt.

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select CATIA from the Application Menu:



3. Select the appropriate option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following Cimatron/CATIA options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
```

3. Select the appropriate option.

Each of these options will be explained in a chapter of this section.

About CATIA

CATIA is a very popular CAD/CAM/CAE system, developed by Dassault Systemes and is marketed worldwide and promoted by IBM.

CATIA stands for **C**omputer-Graphics **A**ided **T**hree Dimensional **I**nteractive **A**pplication.

CATIA is used by the world's leading car manufacturers such as Audi, BMW, DAEWOO, Ferrari, FIAT, HONDA, CHRYSLER, MERCEDES, Peugeot, PORSCHE, Rolls Royce, Rover, ŠKODA, VW, and others. ❑

Chapter 29
The Read-CATIA Utility

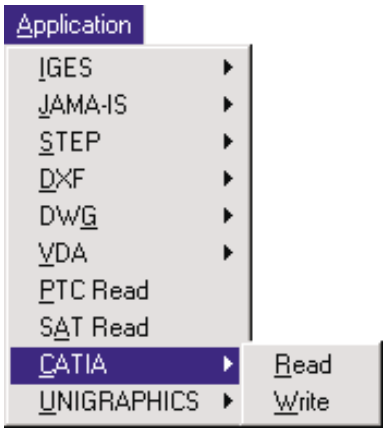
The **Cimatron** Read-CATIA Utility inputs a model file created by CATIA versions 3 and 4 and converts it to a **Cimatron** part file (**.pfm**).

Running Read-CATIA

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select CATIA from the Application Menu: The following options are displayed:



3. Select the **CATIA, Read** option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following CATIA options are available:

```
1  ->  Read  option
2  ->  Write  option
3  ->  Parameters  File  option
```

3. Select the **Read** option.

Running Modes

The **Read-CATIA** Utility may be run interactively, or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or:In **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

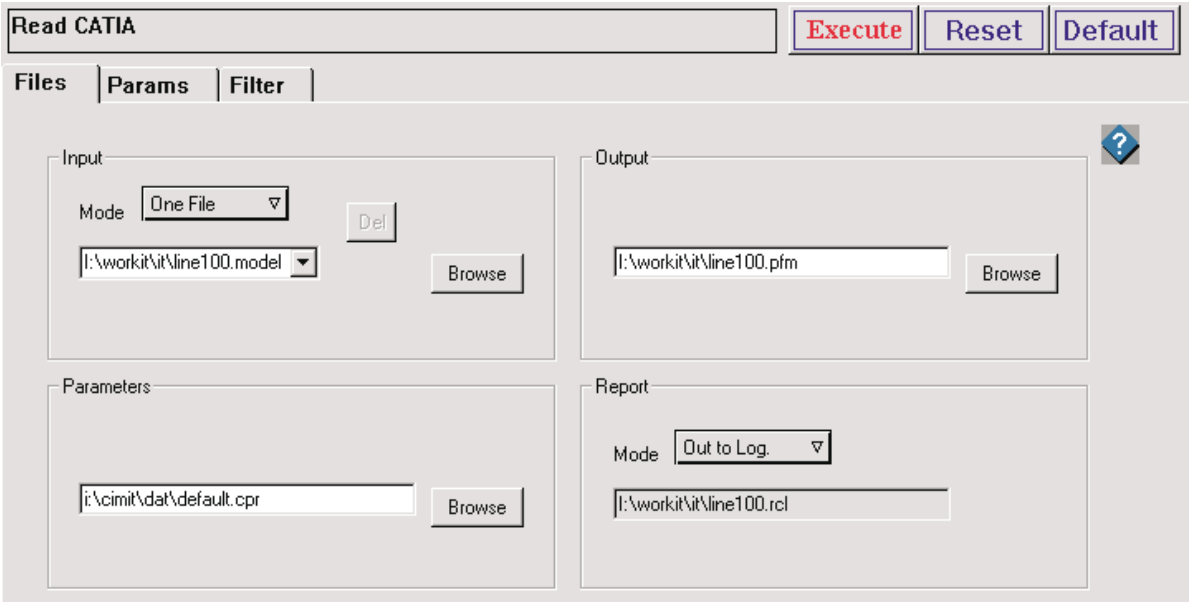
1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file as containing one file name per line.

Notes:

- Each part file created will be in the same directory and have the same name as the CATIA file, but will have the extension **.pfm** instead of the CATIA extension.
- The CATIA file may be located in the current or other directories. Enter the full path name of the CATIA file if it is not located in the current directory.
- In **Batch Mode Without Backup**, if a file already exists with the same name, including the .**pfm** extension, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and **.pfm** files already exist with the name specified in the list, the old **.pfm** file will be copied to a backup file with the extension **.bac**. Then a new **.pfm** file will be created from the CATIA file.

Graphic Interaction with Read-CATIA

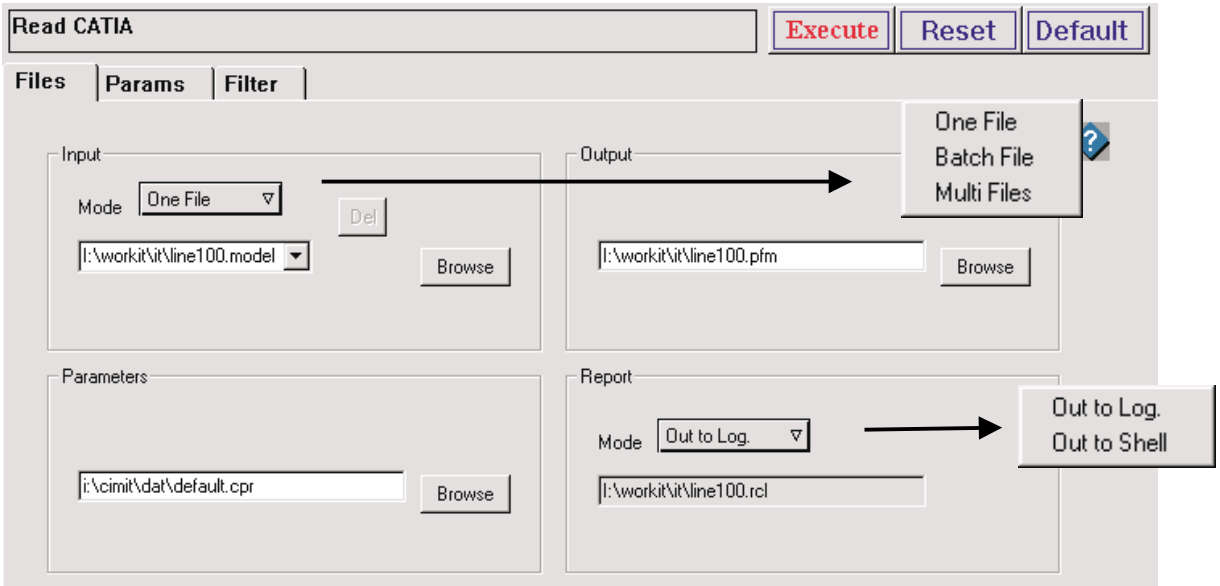


How To:

1. Set the parameters.
2. Select a file in the Input area using the Browse button.
3. Press the Execute button.

Files

This dialog defines the Input, Ouput, Report and Parameter files associated with the translation.



Input

The Read operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input CATIA extension file (the file to be converted to a Cimatron **.pfm**).

The available Input Mode options are:

- ONE FILE — Run the Read operation interactively, one file at a time.
- BATCH FILE — Run the Read operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option
- MULTI FILES — Run the Read operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.pfm** file is identical to the prefix of the input CATIA extension file.

   Choose a different **.pfm** file, or using the browse button, select an existing **.pfm** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the CATIA parameter file is **default.cpr.** You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the Read data interface.
- Insert data required by the Read data interface.
- Influence the behavior of the Read data interface by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read CATIA translation data **.rcl** file is identical to the prefix of the input CATIA extension file.

   Choose a different **.rcl** file, or using the browse button, select an existing **.rcl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

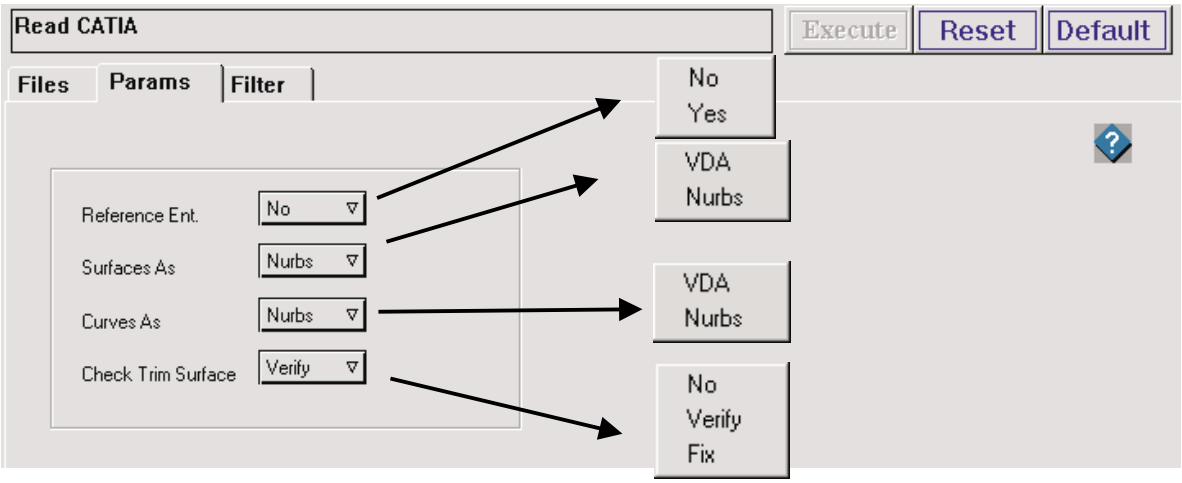
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.

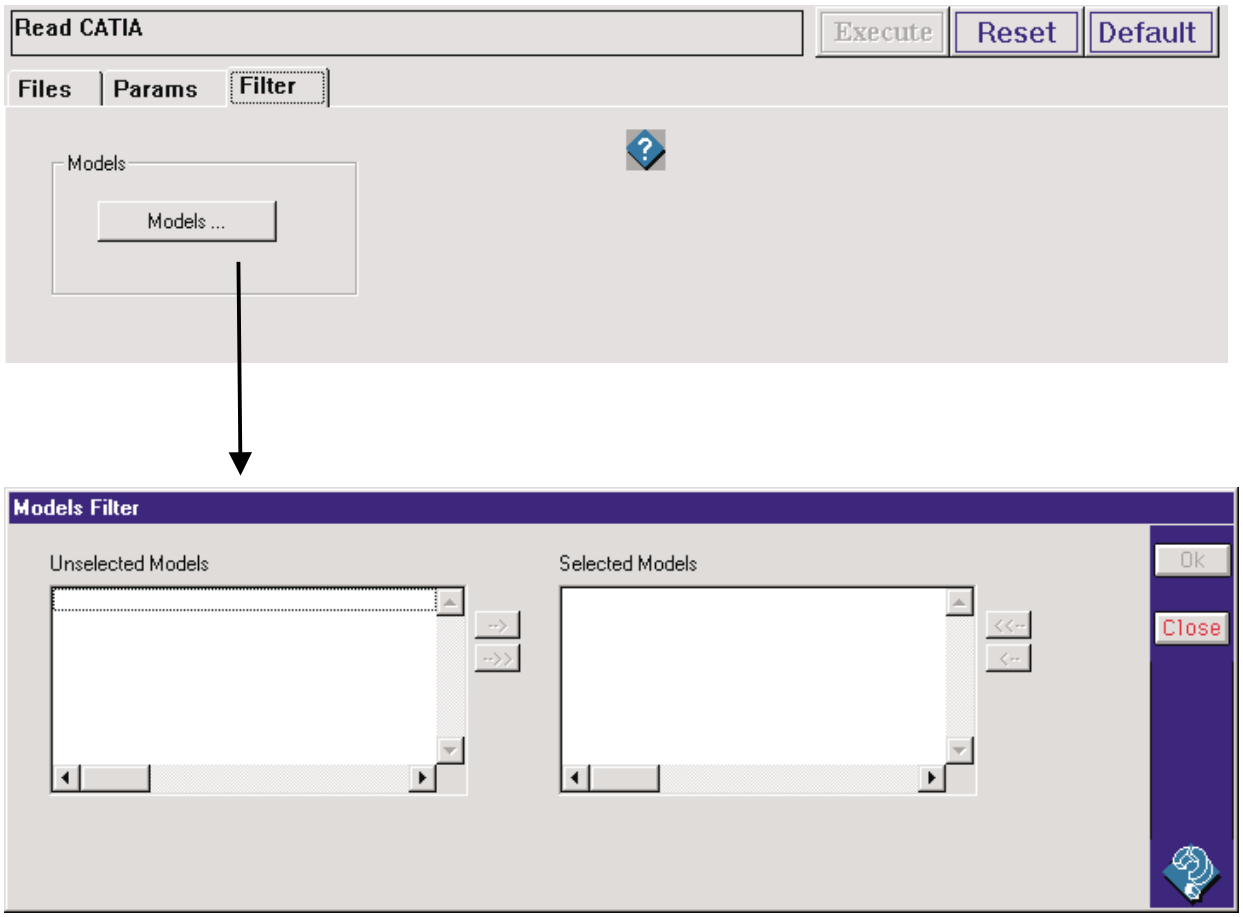


How To:

1. **Reference Ent.**: Choose whether or not to translate all the reference entities in the CATIA file (faces, surfaces, etc..).
2. **Surfaces As**: Read CATIA surfaces as VDA and Nurbs surfaces.
3. **Curves As**: Read CATIA curves as VDA and Nurbs curves.
4. **Check Trim Surface**: Mode of fixing trimmed surfaces (NO, Verify or Fix).

Filter

This dialog defines the filter parameters used during the file translation.



How To:

1. **Models:** Select the models to be included in the translation.

Textual Interaction with Read-CATIA

The following section describes the interaction upon invoking **Read-CATIA**.

Cimatron (version 11.0) CATIA (version 4.0)
==========

Available options :
1 -> Read option
2 -> Parameters file option
Enter option (0 = exit)

Select an option.

CATIA (version 4.0) - Cimatron (version 11.0)
==========

Enter CATIA parameters file name (<CR> = default) :

Available modes:

1 -> Interactive mode
2 -> Batch mode with backup
3 -> Batch mode without backup

The mode menu will appear.

Enter mode (0 = exit)

Type the number of the appropriate operation mode.

To exit the program at this point, type 0 (zero).

If **Interactive mode** is selected:

Enter CATIA file name (<CR>=mode menu)

Type the name of the CATIA file to be converted. To exit the program at this point, press <CR>.

Enter part file name (<CR>=<file_name>) :

Type a name for the part file. To use the name displayed, press <CR>.
If the specified file already exists, the following message appears:

File <file_name>.pfm already exists
Replace existing file ? (Y/N)

To write over the existing file, type **y**. To type a different name for the file, type **n**.

CATIA (version 4.0) - Cimatron (version 11.0)
==========

Setting up part file file_name ...

The new part file will be saved with the **.pfm** extension.

Translating entities from CATIA to Cimatron :
End of entities translation
Saving part <file_name>

Entity Mapping

Table 29-1: Entity Mapping from CATIA to Cimatron

Entity	CATIA Primary type, CATIA secondary type	Cimatron equivalent
Point	1,1	Point
Line	2,1	Line
Polynomial curve	3,1	VDA curve or curve
B-spline polynomial curve	3,2	VDA curve
Polynomial surface	5,1	VDA surface or surface
B-spline polynomial surface	5,2	VDA surface
Face (*)	6,1	Trimmed surface or planar face
Volume (*)	7,1	Trimmed surfaces
Edge	12,1	Element of a contour buffer
Skin (*)	13,1	Trimmed surfaces
Exact solid (*)	17,2	Trimmed surfaces
Circle	20,1	Circle
Ellipse	21,1	Ellipse
Parabola	22,1	Parabola
Hyperbola	23,1	Hyperbola
Composite curve	24,1	Composite curve
SPACE ditto (*)	28,1	Instance of master

Notes:

- All CATIA Solid entities are imported into Cimatron as trimmed surfaces.
- Entities with NS (No Show) will be translated as BLANK

* Short CATIA entity descriptions:

1. FACE - trimmed surface, the underlying surface being either a plane or a polynomial surface.
2. VOLUME - Portion of space bounded by a set of joined faces. These faces are grouped into one or more closed domains.
3. SKIN - SPACE set of joined (composite) faces, which can have a thickness, associated
4. EXACT SOLID - Solid whose graphic representation corresponds to a set of faces, curves, and surfaces which are exactly defined. The representation is related to a topological model which allows access to the faces, curves, and surfaces defining the solid.
5. DITTO - SPACE detail occurrence which can be positioned anywhere within a workspace. It is a single entity and is manipulated as such. An element of ditto cannot be modified, the detail itself must be modified as the ditto elements are only images of the detail elements.

Usage Envelope

1. In cases where a CATIA file contains more than one model, only the first model can be read.
2. In cases where a CATIA model contains several layers, they will be read into Cimatron in the same level numbers but will not be presented in ascending order.
3. All CATIA models will be read into Cimatron as surfaces.
4. The product does not support drawings. ❑

Chapter 30
The Write-CATIA Utility

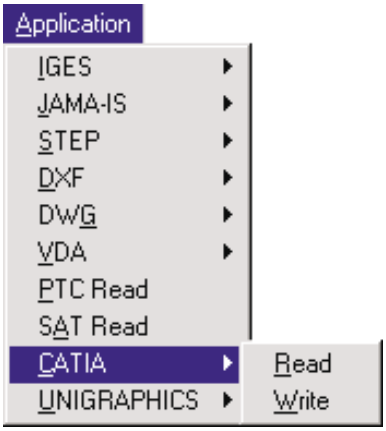
The **Cimatron** Write-CATIA Utility converts **Cimatron** part files (**.pfm**) to files in the CATIA format (in the **.model** or **.exp** extensions).

Running Write-CATIA

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select CATIA from the Application Menu: The following options are displayed:



3. Select the **CATIA, Write** option.

Textual

How To:

1. Follow the instructions in the Textual column of the table on page 1-1.
2. The following CATIA options are available:

   1 -> Read option
   2 -> Write option
   3 -> Parameters File option

3. Select the Write option.

Running Modes

The **Write-CATIA** Utility may be run interactively, or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or: In **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

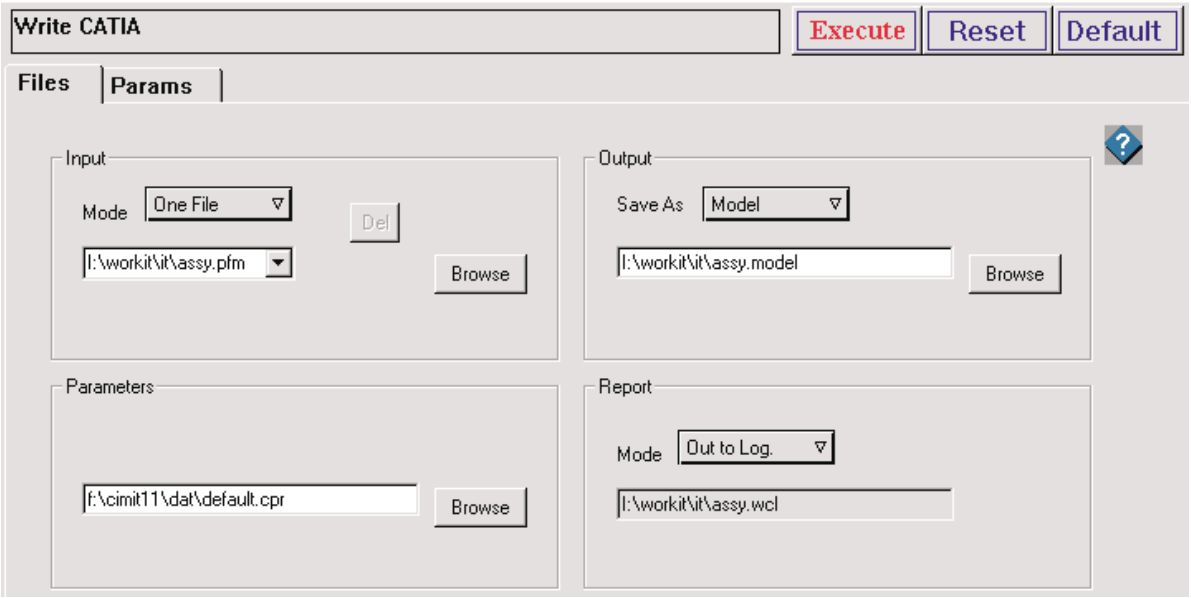
1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file as containing one file name per line.

Notes:

- Each CATIA file created will be in the same directory and have the same name as the Cimatron part file, but will have the CATIA extension instead of the **.pfm** extension.
- Both the part file and the generated CATIA file may be located in the current or other directories. Enter the full path name of the file if it is not located in the current directory.
- In **Batch Mode Without Backup**, if a file already exists with the same name, including the CATIA extension, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and the CATIA files already exist with the name specified in the list, the old CATIA file will be copied to a backup file with the extension **.bac**. Then a new CATIA file will be created from the **.pfm** file.

Graphic Interaction with Write-CATIA

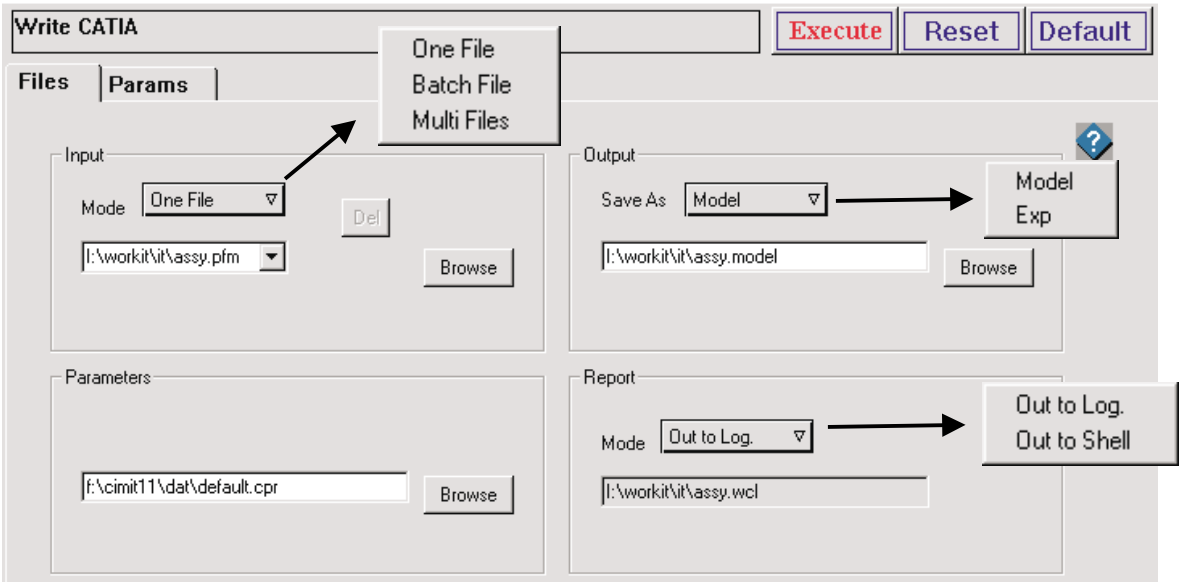


How To:

1. Set the parameters.
2. Select a file in the Input area using the Browse button.
3. Press the Execute button.

Files

This dialog defines the Input, Ouput, Report and Parameter files associated with the translation.



Input

The Read operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input CATIA extension file (the file to be converted to a Cimatron **.pfm**).

The available Input Mode options are:

- ONE FILE — Run the Read operation interactively, one file at a time.
- BATCH FILE — Run the Read operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option
- MULTI FILES — Run the Read operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output CATIA extension file is identical to the prefix of the input **.pfm** file.

   Choose a different CATIA extension file, or using the browse button, select an existing CATIA extension file. In the latter case, you will be prompted whether or not to overwrite the existing file.
2. Select the extension for the CATIA output file: **.model** or **.exp**.

Parameters

How To:

1. By default, the name of the CATIA parameter file is **default.cpr**. You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the Write data interface.
- Insert data required by the Write data interface.
- Influence the behavior of the Write data interface by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read CATIA translation data **.wcl** file is identical to the prefix of the input CATIA extension file.

   Choose a different **.wcl** file, or using the browse button, select an existing **.wcl** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

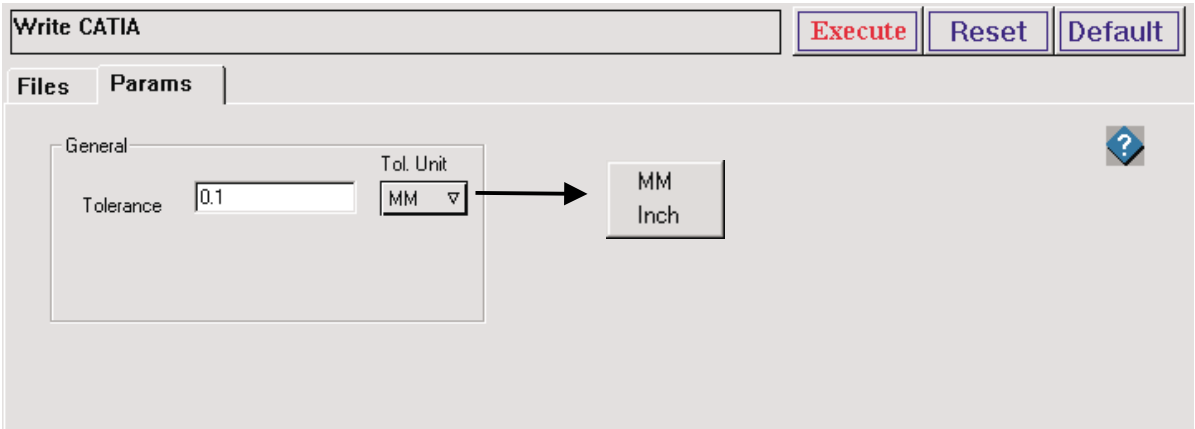
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **Tolerance**: Set the tolerance.
2. **Tol. Unit**: Set the units for the tolerance in MM or Inches.

Textual Interaction with Write-CATIA

The following section describes the interaction upon invoking **Write-CATIA**.

Cimatron (version 11.0) CATIA (version 4.0)
==========

Available options :
1 -> Read option
2 -> Write option
3 -> Parameters file option
Enter option (0 = exit)

Select an option.

Cimatron (version 11.0) CATIA (version 4.0)
==========

Enter CATIA parameters file name (<CR> = default) :

Available modes:

1 -> Interactive mode
2 -> Batch mode

The mode menu will appear.

Enter mode (0 = exit)

Type the number of the appropriate operation mode.

To exit the program at this point, type 0 (zero).

If **Interactive mode** is selected:

Enter part file name (<CR>=mode menu) :

Type a name for the part file. To use the name displayed, press <CR>.

Enter CATIA file name (<CR>=<file_name>)

Type the name of the CATIA file to be converted. To exit the program at this point, press <CR>.
If the specified file already exists, the following message appears:

File <file_name> already exists
Replace existing file ? (Y/N)

To write over the existing file, type **y**. To type a different name for the file, type **n**.

Save CATIA log file <Y/N> ? (<CR> = Y)

Choose whether or not to save the CATIA log file.

Setting up part file <file_name> ...

The new part file will be saved with the **.pfm** extension.

Translating entities from Cimatron to CATIA :
End of entities translation
Saving file <CATIA file_name>
Saving file <CATIA log file_name>

Entity Mapping

Table 30-1: Entity Mapping from Cimatron to CATIA

Cimatron (class,typ)	Catia (typ,subtyp)
point(0,1)	point (1)
line(1,1)	line(2)
circle(1,2)	circle(20)
conic(1,3)	ellipse(21) parabola(22) or hyperbola(23)
2d spline(1,4)	curve(3) D
? (1,5)	curve(3) A
project(1,6)	curve(3) A
offset(1,7)	curve(3) A
2d compost(1,15)	set of lines/arcs/conics/curves
helix(2,2)	curve(3) A
3d cubic spline(2,3)	curve(3) D
3d spline(3,3)	curve(3) A
3d compost(2,15)	set of lines/arcs/conics/curves
p drive surface(3,2)	surface(5) A
ruled surface(3,3)	surface(5) A
revol surface(3,4)	surface(5) A
iges surface(3,5)	surface(5) A
vda surface(3,6)	surface(5) D
b-spline surface(3,11)	surface(5) D
bezier mesh(3,12)	surface(5) D
plface(4,1)	face(6) lying on a surface(5) D
trimsrf(4,2)	face(6) lying on a surface(5) D
instan(13,1)	space ditto(28)
master(14,1)	space detail(1000,4)
group(14,9)	space detail(1000,4)

A : always approximated to cubic bezier
D : approximated depending of the parameters (knot vector)

Solids are translated as trimmed surfaces.

❑

Chapter 31
The CATIA Utilities Parameter File

The CATIA Utilities parameter file allows the CATIA utilities user to:

- Customize the interaction with the utilities.
- Insert data required by the utilities.

This chapter contains a general description of the creation, format and use of the parameter file. Further information on the **Read-CATIA** parameters can be found in Chapter 29, **The Read-CATIA Utility** chapter. Further information on the **Write-CATIA** parameters can be found in Chapter 30, **The Write-CATIA Utility** chapter.

A description of the **Parameter File Check** Utility is given at the end of this chapter.

Parameter File Name

The format of the parameter file name is **filename.cpr**. There are no limitations on the name, however the extension must be **.cpr**. For example, the name of the default parameter file is **default.cpr**.

Parameter File Creation

The **Cimatron** CATIA parameter file is created by the user. Since the file is in ASCII (American Standard Code for Information Interchange) format, it can be easily created using any text editor. The **Cimatron** original default parameter file (**default.cpr**), located in the **<root_cad>\dat** directory, can be used as a template and guide for the user.

Parameter File Format and Rules

The parameter file is an ASCII file. The file contains statements which consist of predefined keywords and their parameter values. Following are some general rules of the parameter file format:

- The order of the lines is not important.
- Letter case is not important except when specifying **Cimatron** files, in which case the **Cimatron** file naming rules must be used.
- Each line starts with a keyword.
- If the same keyword appears more than once, and the statements can be accumulated, they will be; otherwise the last statement encountered will be used.
- A line that starts with ## is a comment line.
- Empty lines are ignored.
- Maximum of 50 words are allowed in one statement.
- To continue the arguments of a statement onto an additional line, finish the current line with \<CR> (backslash followed by carriage return) and continue on to the next line.
- The maximum number of characters allowed in one statement is 256.

Read-CATIA Parameters

The parameter file (**.cpr**) contains parameters that control the operation mode, determine the interaction when running **Read-CATIA** and provide information needed by the utility. These parameters will be described here.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP |<NOBACKUP>]}

  The parameter R-MODE selects the mode of operation. **Read-CATIA** may be run interactively or in batch mode. Optional backup is available with batch mode.

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. User is prompted for name of each CATIA file to be converted.
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the CATIA files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.

BACKUP Option used in batch mode. If a **.pfm** file already exists with the name specified in the batch file, it will be copied to a backup file with the extension **.bac.**

NOBACKUP This is a system default option used in batch mode. If a part file (**.pfm**) with the name specified in the list already exists the new part file will replace it.

Defaults: ASK, NOBACKUP

- SELECT R-SURFACES-AS {VDA | <NURBS>}

  Mode of reading CATIA faces.

  VDA Surfaces are read as VDA surfaces.

  NURBS Surfaces are read as NURBS surfaces.

  Default : NURBS

- SELECT R-CURVES-AS {VDA | <NURBS>}

  Mode of reading CATIA faces.

  VDA Curves are read as VDA curves.

  NURBS Curves are read as NURBS curves.

  Default : NURBS

- SELECT R-FIX-TRMSRF {NO | <VERIFY> | FIX}

  Mode of fixing trimmed surfaces.

  Default: VERIFY

- SELECT R-UCS-DISPLAY {<NO> | YES}

  Mode of converting UCS.

  Default: NO

- SELECT R-SPLIT-SRF {<NO> | YES}

  Split a surface on each point when the surface is not smooth.
  This option checks every surface for smoothness. If a break point is found, the surface is split at the break point and two surfaces created as a result.

  NO Do not split on break points.

  YES Split surfaces on break points.

  Default: NO

- SELECT R-CONVERT-NO-BOUND-SRF {<NO> | YES}

  Convert surfaces that have no bounds.

  Default: NO

- SELECT R-MODELS {<0> | MODEL NUMBERS}

  Filter models.

  Default: 0

- SELECT R-REFINE-TRMSRF-TOL {approximation-tolerance(0.0-1.0) | <0.0>}

  Refine trimmed surface boundaries when required (TOL > 0 and TOL < 1). The default option (0) parameter will not be used in the data interface.

  If the value is between 0 and 1, this will be used as the tolerance for the Refine Trmsrf algorithm which will be forced on the resulting trimmed surfaces after the data interface process is completed.

  If you set the value <0 or >1, the application will assume that 0.0 was used and the parameter will be ignored.

  Default: 0.0

Write-CATIA Parameters

The parameter file (**.cpr**) contains parameters that control the operation mode, determine the interaction when running **Write-CATIA** and provide information needed by the utility. These parameters will be described here.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SET W-TOL {approximation-tolerance (0.001 - 100) | <0.1>}

  The tolerance for writing the approximation.

  Default: 0.1

- SET W-TOL-UNITS {<MM> | INCH}

  The units of the tolerance for writing the approximation.

  Default: MM

Parameter File Check

The CATIA Utilities provides an option to check the format and statement syntax of the parameter file.

To check a file, use option **<2> PARAMETER FILE CHECK**, from the CATIA Utilities menu.

If no errors are found in the parameter file, the following message will appear:

PARAMETER FILE WAS SUCCESSFULLY PARSED ❑

Section X

Unigraphics Utilities

Chapter 32
Introduction to the Unigraphics Utilities

The **Cimatron** Read-Unigraphics Utility inputs a model file created by Unigraphics Version 11 *or later* and converts it to a Cimatron part file (**.pfm**).

The **Cimatron** Read-Unigraphics Utility is an independent utility that does not require a Unigraphics license. It will only handle geometric Unigraphics data form.

The following functions are included in the **Cimatron** Unigraphics Utilities:

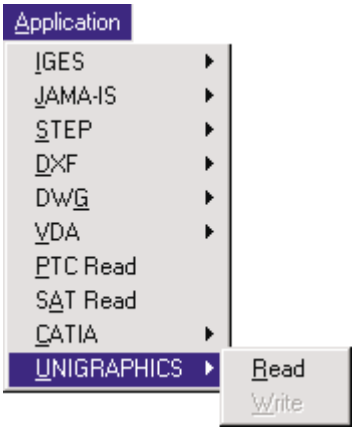
Read-Unigraphics	Convert files that are in UNIGRAPHICS (**.prt**) format to files in **Cimatron** part file (**.pfm**) format.
PARAMETER FILE CHECK	Verify a Unigraphics parameter file (**.upr**).

Running Unigraphics

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select UNIGRAPHICS from the Application Menu:



3. Select the appropriate option.

About Unigraphics

Unigraphics is the leading CAD/CAM/CAE software solution for enterprise-wide design-through manufacturing productivity. Unigraphics is an advanced software system for companies who produce complex products such as cars, airplanes, medical implant devices and machine tools. It is also used extensively worldwide to create a wide variety of items we use every day, such as boats, bicycles, toys, telephones, coffee makers and golf clubs. Unigraphics provides companies a complete digital product model for design, engineering, analysis and manufacturing. ❑

Chapter 33
The Read-Unigraphics Utility

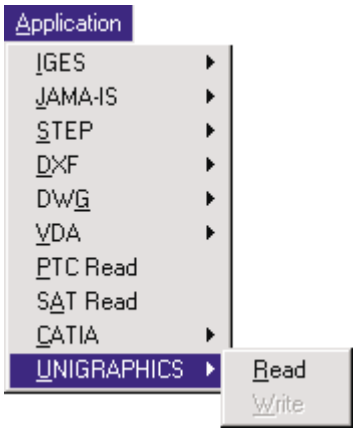
The **Cimatron** Read-Unigraphics Utility inputs a model file created by Unigraphics Version 11 *or later* and converts it to a Cimatron part file (**.pfm**).

Running Read-Unigraphics

Graphic

How To:

1. Follow the instructions in the Graphic column of the table on page 1-1.
2. Select Unigraphics from the Application Menu: The following options are displayed:



3. Select the Unigraphics, Read option.

Running Modes

The **Read-Unigraphics** Utility may be run interactively, or in batch mode.

Interactive

In **INTERACTIVE** mode:

How To:

1. Select/type the name of each file, one at a time.
2. Type a different name for the converted part file, if desired.

Or: In **Multi** mode:

1. Select multiple files.

Batch

In **BATCH** mode, either with or without a backup file:

How To:

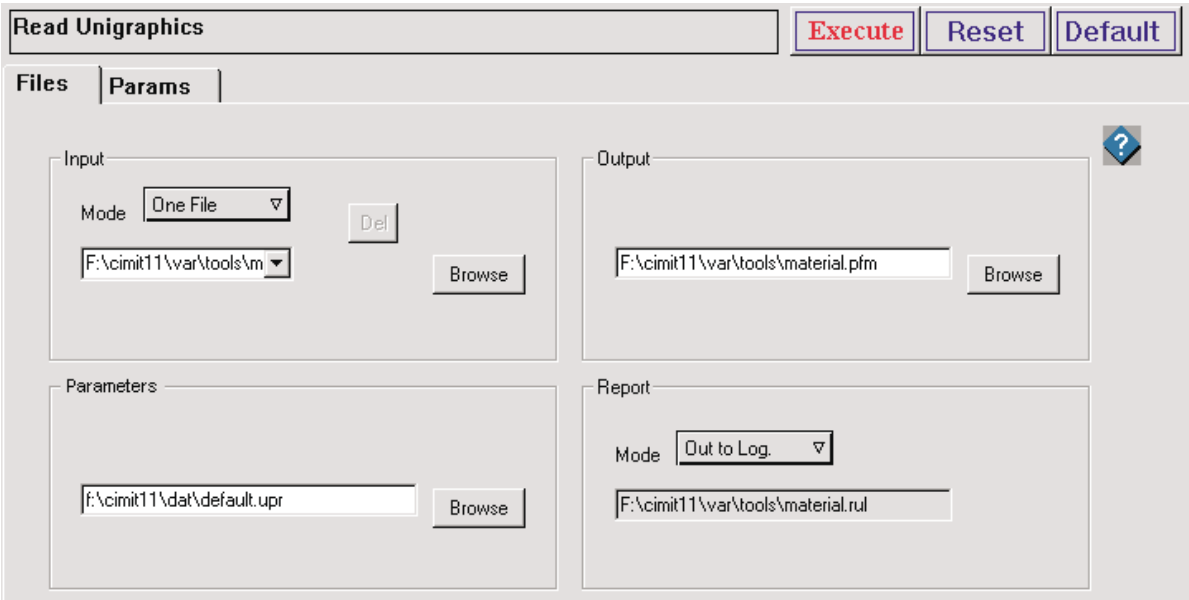
1. Read file names from a batch file.
2. Convert each file listed in the batch file.

In **BATCH** mode, a batch file (list file) must be prepared before the conversion program is run. There are no restrictions on the name of the batch file. Use the editor to create a batch file as containing one file name per line.

Notes:

- Each part file created will be in the same directory and have the same name as the Unigraphics file, but will have the extension **.pfm** instead of the Unigraphics extension.
- The Unigraphics file may be located in the current or other directories. Enter the full path name of the Unigraphics file if it is not located in the current directory.
- In **Batch Mode Without Backup**, if a file already exists with the same name, including the **.pfm** extension, the file created with the program will replace it.
- If **Batch Mode With Backup** was selected, and **.pfm** files already exist with the name specified in the list, the old **.pfm** file will be copied to a backup file with the extension **.bac**. Then a new **.pfm** file will be created from the Unigraphics file.

Graphic Interaction with Read-Unigraphics

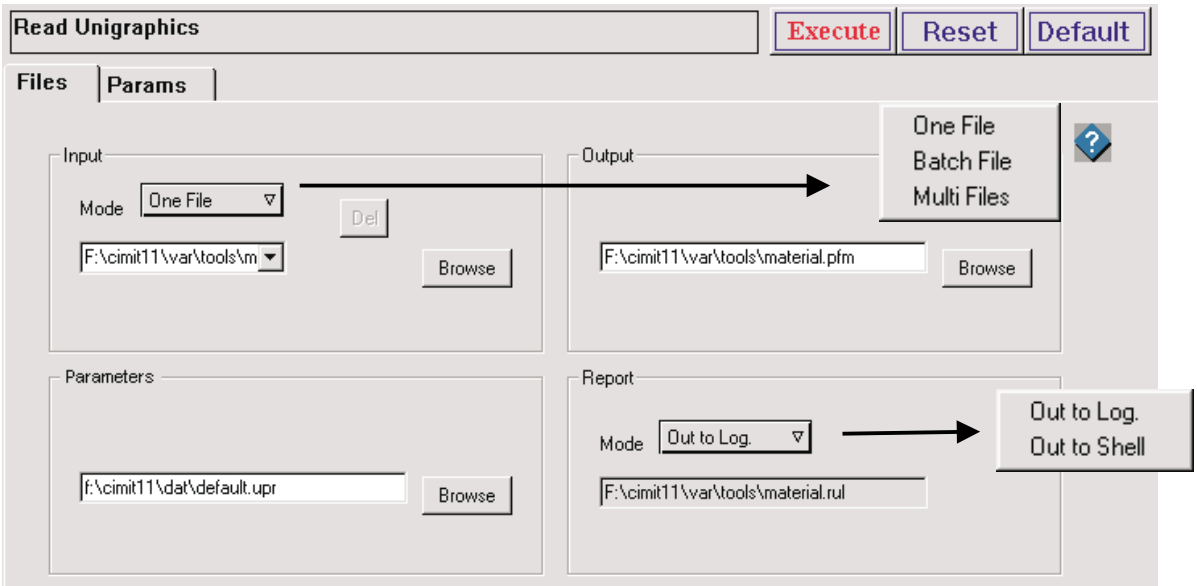


How To:

1. Set the parameters.
2. Select a file in the Input area using the Browse button.
3. Press the Execute button.

Files

This dialog defines the Input, Ouput, Report and Parameter files associated with the translation.



Input

The Read operation may be run in either **interactive** or **batch** mode.

How To:

1. Select the required Input Mode: One File, Batch File or Multi Files.
2. Using the Browse button, choose the input Unigraphics extension file (the file to be converted to a Cimatron **.pfm**).

The available Input Mode options are:

- ONE FILE — Run the Read operation interactively, one file at a time.
- BATCH FILE — Run the Read operation in BATCH mode. The mode menu will not appear on the screen. The names of the files to be converted are taken from a batch file that you must create. You will be prompted for the name of the batch file.

  Read the part file names from:

  A list in a batch file.

  FILTER PARAMETER is not available in this option
- MULTI FILES — Run the Read operation interactively, multiple files at a time.

Output

How To:

1. By default, the prefix of the output **.pfm** file is identical to the prefix of the input Unigraphics extension file.

   Choose a different **.pfm** file, or using the browse button, select an existing **.pfm** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

Parameters

How To:

1. By default, the name of the Unigraphics parameter file is **default.upr**. You may use this name or select an existing parameter file name using the Browse button.

The **parameter file** enables you to:

- Customize the interaction with the Read data interface.
- Insert data required by the Read data interface.
- Influence the behavior of the Read data interface by means of user-filters.

Report

Translation data can either be displayed on the screen or saved to a log file.

This log file contains information such as:

- User parameter values.
- Statistical information on the entities in the file being converted.

How To:

1. Select the required Report Mode: OUT TO LOG or OUT TO SHELL.
2. By default, the prefix of the Read Unigraphics translation data **.rul** file is identical to the prefix of the input Unigraphics extension file.

   Choose a different **.rul** file, or using the browse button, select an existing **.rul** file. In the latter case, you will be prompted whether or not to overwrite the existing file.

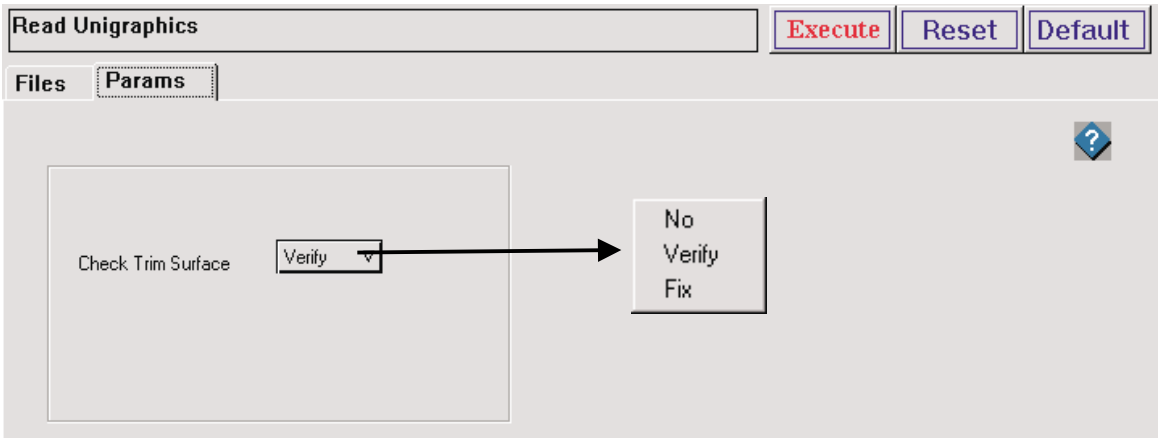
The available Report Mode options are:

- OUT TO LOG — All translation data will be saved to a log file.
- OUT TO SHELL — Translation data will be displayed on screen.

  The screen will display information about the class and type contained in the part file, and will also list the entities that were not translated.

Params

This dialog defines the parameters used during the file translation.



How To:

1. **Check Trim Surface**: Mode of fixing trimmed surfaces (NO, Verify or Fix).

Entity Mapping

Table 33-1: Entity Mapping from Unigraphics to Cimatron

Entity	Unigraphics type	Cimatron equivalent
Point	2	Point
Line	3	Line
Arc	5	Arc
Conic	6	Conic
Spline	9	3D Spline
Solid	70	Trimsrf

❑

Chapter 34
The Unigraphics Utilities Parameter File

The Unigraphics Utilities parameter file allows the Unigraphics utilities user to:

- Customize the interaction with the utilities.
- Insert data required by the utilities.

This chapter contains a general description of the creation, format and use of the parameter file. Further information on the **Read-Unigraphics** parameters can be found in Chapter 33, **The Read-Unigraphics Utility** chapter.

Parameter File Name

The format of the parameter file name is **filename.upr**. There are no limitations on the name, however the extension must be **.upr**. For example, the name of the default parameter file is **default.upr**.

Parameter File Creation

The **Cimatron** Unigraphics parameter file is created by the user. Since the file is in ASCII (American Standard Code for Information Interchange) format, it can be easily created using any text editor. The **Cimatron** original default parameter file (**default.upr**), located in the **<root_cad>\dat** directory, can be used as a template and guide for the user.

Parameter File Format and Rules

The parameter file is an ASCII file. The file contains statements which consist of predefined keywords and their parameter values. Following are some general rules of the parameter file format:

- The order of the lines is not important.
- Letter case is not important except when specifying **Cimatron** files, in which case the **Cimatron** file naming rules must be used.
- Each line starts with a keyword.
- If the same keyword appears more than once, and the statements can be accumulated, they will be; otherwise the last statement encountered will be used.
- A line that starts with ## is a comment line.
- Empty lines are ignored.
- Maximum of 50 words are allowed in one statement.
- To continue the arguments of a statement onto an additional line, finish the current line with \<CR> (backslash followed by carriage return) and continue on to the next line.
- The maximum number of characters allowed in one statement is 256.

Read-Unigraphics Parameters

The parameter file (**.upr**) contains parameters that control the operation mode, determine the interaction when running **Read-Unigraphics** and provide information needed by the utility. These parameters will be described here.

The conventions used to describe the parameters are given in the preface.

Operation Mode Parameters

- SELECT R-MODE {<ASK> | INTER | BATCH} [BACKUP |<NOBACKUP>]}

  The parameter R-MODE selects the mode of operation. **Read-Unigraphics** may be run interactively or in batch mode. Optional backup is available with batch mode.

ASK	Mode menu will appear. User is prompted for operation mode.
INTER	Program will run in interactive mode. The mode menu will not appear on the screen. User is prompted for name of each Unigraphics file to be converted.
BATCH	Program will run in batch mode. The mode menu will not appear on the screen. The names of the Unigraphics files to be converted are taken from a batch file that must be created by the user. The user is prompted for the name of the batch file.

BACKUP Option used in batch mode. If a **.pfm** file already exists with the name specified in the batch file, it will be copied to a backup file with the extension **.bac.**

NOBACKUP This is a system default option used in batch mode. If a part file (**.pfm**) with the name specified in the list already exists the new part file will replace it.

Defaults: ASK, NOBACKUP

- SELECT R-FIX-TRMSRF {NO | <VERIFY> | FIX}

  Mode of fixing trimmed surfaces.

  Default: VERIFY

-

Section X

Appendices

Appendix A
Cimatron File Name Extensions

Cimatron has a number of file types associated with Data Interfaces. Each specific file type has a unique extension name.

Cimatron Extensions

.pfm	**Cimatron** part file.
.bac	**.pfm** backup file.
.pfr	**.pfm** file information.

DXF Extensions

.dxf	DXF file.
.rdl	**Read-DXF** Log File containing translation data.
.wdl	**Write-DXF** Log File containing translation data.
.dpr	DXF Parameter file.
.drt	DXF file information.

DWG Extensions

.dwg	DWG file.
.rgl	**Read-DWG** Log File containing translation data.
.wgl	**Write-DWG** Log File containing translation data.
.dpr	DWG Parameter file.
.drt	DWG file information.

IGES Extensions

.igs	IGES file.
.ril	**Read-IGES** Log File containing translation data.
.wil	**Write-IGES** Log File containing translation data.
.ris	**Read-IGES** Start Section Information File.
.wis	**Write-IGES** Start Section Information File.
.ipr	IGES Parameter file.

PTC Extensions

.neu	PTC neutral file.
.rpl	**Read-PTC** Log File containing translation data.
.ppr	PTC Parameter file.

VDA Extensions

.vda	VDA file.
.rvl	**Read-VDA** Log File containing translation data.
.wvl	**Write-VDA** Log File containing translation data.
.vpr	VDA Parameter file.

Jama-Is Extensions

.rjl	**Read-JAMA** Log File containing translation data.
.wjl	**Write-JAMA** Log File containing translation data.
.jpr	JAMA-Is Parameter file.
.jrt	JAMA-Is file information.

STEP Extensions

.stp (or .step)	STEP file.

CATIA Extensions

without any extension **.model** **.mdl** **.cat** **.mod** **.dlv3** **.dlv4** **.dlv** **.exp**	CATIA extensions.
.rcl	**Read-CATIA** Log File containing translation data.
.wcl	**Write-CATIA** Log File containing translation data.
.cpr	CATIA Parameter file.

Unigraphics Extensions

.prt	Unigraphics file.
.rul	**Read-Unigraphics** Log File containing translation data.
.upr	Unigraphics Parameter file.

❑

Appendix B
DWG Group Codes

The specific assignment of group codes depends on the item being described in the file. However, the type of the value this group supplies is derived from the group code in the following way:

Table B-1: Group Code Ranges

Group Code Range	Following Value	Group Code Range	Following Value
0 - 9	String	210- 239	Floating-point
10 - 59	Floating-point	999	Comment (string)
60 - 79	Integer	1000 - 1009	String
140 - 147	Floating-point	1010 - 1059	Floating-point
170 - 175	Integer	1060 - 1-79	Integer

Group Codes

Group codes are used both to indicate the type of the value of the group, as explained earlier, and to indicate the general use of the group. The specific function of the group code depends on the actual variable, table item, or entity description. The following table indicates the general use of groups, noting as “fixed” any that always have the same function.

Table B-2: General Use of Group Codes

Group Code	Value Type
0	Identifies the start of an entity, table entry, or file separator. The type of entity is given by the text value that follows this group.
1	The primary text value for an entity.
2	A name: Attribute tag, Block name, and so on. Also used to identify a DWG section or table name.
3 - 4	Other textual or name values.
5	Entity handle expressed as a hexadecimal string (fixed).
6	Line type name (fixed).
7	Text style name (fixed).
8	Layer name (fixed).
9	Variable name identifier (used only in HEADER section of the DWG file).
10 - 18	X coordinates.
20 - 28	Y coordinates. 2n values always correspond to 1n values and immediately follow them in the file.
30 - 37	Z coordinates. 3n values always correspond to 1n and 2n values and immediately follow them in the file.
38	The entity’s elevation, if non zero (fixed). Exists only in output from versions prior to 11.0.

Group Code	Value Type
39	The entity's thickness, if non zero (fixed).
40 - 48	Floating point values (text height, scale factors, etc.).
49	Repeated value - multiple 49 groups may appear in one entity for variable length tables (such as the dash lengths in the LTYPE table). A 7x group always appears before the first 49 group to specify the table length.
50 - 58	Angles
62	Color number (fixed).
66	"Entities follow" flag (fixed).
67	Identifies whether entity is in model space or paper space.
68	Identifies whether viewport is on but fully off screen, is not active, or is off.
69	Viewport identification number.
70 - 78	Integer values such as repeat counts, flag bits, or modes.
210, 220, 230	X, Y, and Z coordinates of extrusion direction (fixed).
999	Comments

❑

Index

C

CATIA
About 28-3
Entity Mapping, Read-CATIA 29-10, 30-9
Introduction 28-1
Invoking Read-CATIA textual mode 29-9
Invoking Write-CATIA textual mode 30-8
Parameter File 31-1 - 31-4, 34-1 - 34-3
Parameter File, Read-CATIA 31-2
Parameter File, Write-CATIA 31-4
Read-CATIA 29-1
Running 28-2
Running Read-CATIA 29-1
Running Write-CATIA 30-1
Write-CATIA 30-1

D

Data Interfaces Utilities 1-1
DWG
Introduction 2-1
DXF
description of format 2-3 - 2-4
File Information Utility 6-1
Group Codes B-1
How does it work? 2-3
Interacting with Read DXF:Textual 3-11
Interacting with Read-DXF:Graphic 3-4
Interacting with Write-DXF:Graphic 4-4
Interacting with Write-DXF:Textual 4-12
Introduction 2-1
Parameter File 5-1
Parameter File Creation 5-1
Parameter File Format & Rules 5-2
Parameter File Name 5-1
Parameter File Statements & Syntax 5-2
PFM File Information Util. 7-1
Read DXF Running Modes 3-3
Read-DXF 3-1
Running Read DXF 3-2
Running Write DXF 4-2
What is DXF? 2-3
Write DXF Running Modes 4-3
Write-DXF 4-1
DXF Utilities
File Information Utility 14-2
DXF-Read
Attribute Mapping 3-12
Entity Mapping 3-13
Parameters 5-3
DXF-Write
Application Mapping 4-13
Attibute Mapping 4-13
Entity Mapping 4-14
Parameters 5-7
Selective Translation 5-11

F

File name
extensions A-1
filters 11-24

G

Group Codes, DXF B-1

I

IGES
description of format 8-3
Entity Mapping, Read IGES 9-15
Entity Mapping, Write-IGES 10-16
File Format Overview 8-4
File Information Utility 12-1 - 12-3, 13-1 - 13-2

File Structure 8-5

How does it work? 8-3

Introduction 8-1

Invoking Read-IGES 9-14

Invoking Write-IGES 10-15

Parameter File 11-1

PFM File Information Utility 13-1

Read-IGES 9-1

Running 8-2, 21-2

Running Read-IGES 9-2

Running Write-IGES 10-2

Selective Translation, Write-IGES 11-24

What is it? 8-3

Write-IGES 10-1

IGES parameter file 11-1, 11-3, 11-5, 11-7, 11-9, 11-11, 11-13, 11-15, 11-17, 11-19, 11-21, 11-23, 11-25, 11-27, 11-29

IGES-Write

Parameters 11-11

Interacting with Read DXF:Graphic 3-4

Interacting with Read DXF:Textual 3-11

Interacting with Write-DXF:Graphic 4-4

Interacting with Write-DXF:Textual 4-12

Interaction - Graphic 1-2

J

JAMA FILE INFORMATION 21-1

JAMA-IS 22-1

Description of format 21-3

File Structure 22-2

Introduction 21-1

Options 22-1

Running 21-2

Specification 21-3

Japan Automobile Manufacturers Assoc. 21-1

N

Nested parameter file

IGES utilities 11-23

P

parameter file

DXF 11-2, 11-4, 11-6, 11-8, 11-10, 11-12, 11-14, 11-16, 11-18, 11-20, 11-22, 11-24, 11-26, 11-28, 11-30

IGES utilities 11-1, 11-3, 11-5, 11-7, 11-9, 11-11, 11-13, 11-15, 11-17, 11-19, 11-21, 11-23, 11-25, 11-27, 11-29

PARAMETER FILE CHECK 2-1, 21-1

PTC

Introduction 14-1

Invoking Read-PTC 15-9

Parameter File 16-1 - 16-5

Parameters, Read-PTC 16-3

Read-PTC 15-1, 15-3, 15-5, 15-7, 15-9

Running 14-2

Running Read-PTC 15-2

R

Read-DXF 2-1

Read-JAMA 21-1

Read-STEP 24-1

Running Data Interface Utils. 1-1

Running DXF 2-2

Running JAMA-IS 21-2

Running Read DXF 3-2

Running Write DXF 4-2

S

SAT

About SAT 26-1

Introduction 26-1

Invoking Read-SAT textual mode 27-7

Read-SAT 27-1

Running 26-2

Running Read-SAT 27-1

STEP
About 23-3
Entity Mapping, Read-STEP 24-7
Entity Mapping, Write-STEP 25-7
File Structure 23-3
Introduction 23-1
Read-STEP 24-1
Running 23-2
Running Read-STEP 24-1
Running Write-STEP 25-1
What is it? 23-3
Write-STEP 25-1
STEP Utility 23-1

T

Typographical conventions 1-4
in parameter file statements 1-5

U

Unigraphics
About 32-2
Entity Mapping, Read-Unigraphics 33-9
Introduction 32-1
Parameter File 34-1
Parameter File, Read-Unigraphics 34-2
Read-Unigraphics 33-1
Running 32-1
Running Read-Unigraphics 33-1

V

VDA
Interface 17-3
Introduction 17-1
Invoking Read-VDA 18-10
Invoking Write-VDA 19-10
Nested Parameter File 20-8
Parameter File 20-1, 20-3, 20-5, 20-7, 20-9
Parameter File, Read-VDA 20-3
Parameter File, Write-VDA 20-7
Read-VDA 18-1
Running 17-2
Running Read-VDA 18-2
Running Write-VDA 19-2
Write-VDA 19-1
Write-VDA Utility Interaction Statements 20-8
VDA-Read
Parameters 20-3
VDA-Write
Running 19-2

W

Write-DXF 2-1
Write-IGES
parameters 11-11
Write-JAMA 21-1